Evaluation of Intelligent Medical Systems

by

Julian Bernard Tilbury

A thesis submitted to the University of Plymouth

in partial fulfilment for the degree of

Doctor of Philosophy

Department of Communication and Electronic Engineering

Faculty of Technology

September 2002

Evaluation of Intelligent Medical Systems

Julian Bernard Tilbury

Abstract

This thesis presents novel, robust, analytic and algorithmic methods for calculating Bayesian posterior intervals of receiver operating characteristic (ROC) curves and confusion matrices used for the evaluation of intelligent medical systems tested with small amounts of data.

Intelligent medical systems are potentially important in encapsulating rare and valuable medical expertise and making it more widely available. The evaluation of intelligent medical systems must make sure that such systems are safe and cost effective. To ensure systems are safe and perform at expert level they must be tested against human experts. Human experts are rare and busy which often severely restricts the number of test cases that may be used for comparison.

The performance of expert human or machine can be represented objectively by ROC curves or confusion matrices. ROC curves and confusion matrices are complex representations and it is sometimes convenient to summarise them as a single value. In the case of ROC curves, this is given as the Area Under the Curve (AUC), and for confusion matrices by kappa, or weighted kappa statistics. While there is extensive literature on the statistics of ROC curves and confusion matrices they are not applicable to the measurement of intelligent systems when tested with small data samples, particularly when the AUC or kappa statistic is high.

A fundamental Bayesian study has been carried out, and new methods devised, to provide better statistical measures for ROC curves and confusion matrices at low sample sizes. They enable exact Bayesian posterior intervals to be produced for: (1) the individual points on a ROC curve; (2) comparison between matching points on two uncorrelated curves; (3) the AUC of a ROC curve, using both parametric and nonparametric assumptions; (4) the parameters of a parametric ROC curve; and (5) the weight of a weighted confusion matrix.

These new methods have been implemented in software to provide a powerful and accurate tool for developers and evaluators of intelligent medical systems in particular, and to a much wider audience using ROC curves and confusion matrices in general. This should enhance the ability to prove intelligent medical systems safe and effective and should lead to their widespread deployment.

The mathematical and computational methods developed in this thesis should also provide the basis for future research into determination of posterior intervals for other statistics at small sample sizes.

Contents

Figures and Tables

Acknowledgements

I owe **Peter Van-Eetvelt** a great debt for his expert assistance with mathematics. All the ideas in this thesis first took shape as abstract multi-dimensional geometry floating in my mind, then as more concrete chunks of computer pseudo-code. It was only with Peter's help that they have taken rigorous mathematical form. **Levente Toth** also helped in the process of turning my gibberish into mathematics. **Julian Stander** advised on the presentation of the Monte Carol simulation results, which saved me contemplating infinite recursion.

I owe what sanity I have left, and about 7 million cups of coffee, to **John Curnow**. It is surprising how quickly tumbling thoughts neatly organised themselves when I tried to explain them to John. He was also rather good at spotting such incoherence in reports, papers and this thesis.

I owe the rigorous approach to **Emmanuel Ifeachor**. Emmanuel always found a little more to do here, and a little piece to fill in there, which covers a lot of ground, very thoroughly, over the length of a research project.

The literary touch was inspired by **Helen Edmunds**. She does not find three mile long sentences particularly digestible. Many subtle errors were removed thanks to **R John Parsons**' proof reading.

I would also like to thank **Jon Garibaldi**. Firstly for his critique of reports, papers and this thesis, but mainly for the conversations that launched this research.

Finally, I would like to thank my examiners **Paulo Lisboa** and **David Wright**, firstly for spotting a few glitches, but mainly for clearing up a some mysteries that had been nagging for years.

Dedication

To my parents, **Tem** and **Psyche**, who would have liked to see this thesis in their lifetimes.

Author's Declaration

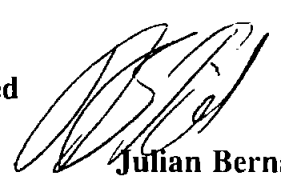
At no time during the registration for the degree of Doctor of Philosophy has the author been registered for any other University award.

This study was financed with the aid of a studentship from the Engineering and Physical Sciences Research Council.

Publications

J. M. Garibaldi, J. Tilbury, E. C. Ifeachor, '*The Validation of a Fuzzy Expert System for Umbilical Cord Acid–Base Analysis*', in Proceedings of the Third International Conference on Neural Networks and Expert Systems in Medicine and Healthcare, World Scientific Publishing, Singapore, ISBN 981–02–3611–5, pp.230–240, 1998.

J. B. Tilbury, P. W. J. Van Eetvelt, J. M. Garibaldi, J. S. H. Curnow, E. C. Ifeachor, '*Receiver Operating Characteristic Analysis for Intelligent Medical Systems – A New Approach for Finding Confidence Intervals*', IEEE Transactions on Biomedical Engineering, Vol. 47, No.7, pp.952–963, 2000.

Signed 

Julian Bernard Tilbury

Date 2002-10-03

1 Introduction

Intelligent medical systems have an important role to play in capturing rare and valuable medical expertise and making it more widely and cheaply available to a society that expects the highest standards of medical care [1]. Such intelligent systems will only reach their potential if they can be shown to be safe, accurate, and cost effective [2][3].

Ever since the first experiments with artificial intelligence techniques in the medical domain, researchers have been identifying the problems of evaluation [4][5] and designing techniques to overcome these problems [6][7]. One problem is obtaining enough expert opinions on example cases to compare with the output of the system [8][9]. Powerful and robust statistical techniques are therefore vital, because the small sample sizes often obtained introduce uncertainly into the results.

The data from the testing of such systems is frequently presented as Receiver Operating Characteristic (ROC) curves [10][11], or as confusion matrices [12]. The diagnostic accuracy of a system can then be quantified in terms of the Area Under the Curve (AUC) for ROC curves, or by the kappa [13][14] statistics for confusion matrices. While confidence intervals are available for both these statistics, they are unreliable where the sample size is small and system performance, as measured by the AUC or kappa, is high. It is therefore important to find robust and accurate statistics for ROC curves and confusion matrices that will work under these circumstances. This thesis takes a novel approach by using Bayesian statistics [15] to produce posterior intervals rather than the confidence intervals of Frequentist statistics.

1.1 ROC Curves

Judging the ability of a system in classifying cases is dependent on knowing the true classification. This problem of defining a standard is non-trivial, and will be examined in Chapter 2.

Assuming an objective standard is established, an intelligent system can be measured against it. The simplest situation is a binary choice about a situation, for example the diagnosis of a single disease. In the simplest instance this is a '*diseased*' or '*healthy*' decision,

but it might be graded into '*definitely diseased*' '*possibly diseased*', '*possibly healthy*', and '*definitely healthy*'. Any number of categories could be used, or the scale could be continuous. For example, an artificial neural network might produce its output as a real number between 0.00 and 1.00.

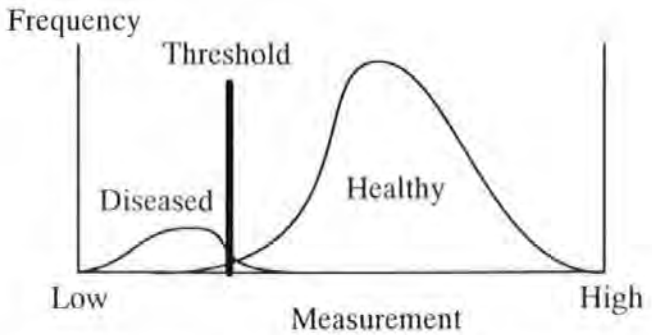


Figure 1.1 The underlying model for ROC curves

ROC curves are based on the model that an intelligent medical system is measuring, either explicitly, or implicitly, a quantity that has a different distribution for the diseased and healthy populations. Figure 1.1 gives a hypothetical example of the relative frequency with which the diseased and healthy cases give different values of the measured quantity. To distinguish between the populations, a threshold is chosen so that cases with a measurement greater than the threshold are labelled as *healthy*, and cases with a measurement lower than the threshold are labelled as *diseased*. Since the two distributions overlap, no threshold value will completely separate them. Table 1.1 shows the 2 by 2 contingency table of the true classification, against its test classification according to the threshold.

		Standard: Diseased	Standard: Healthy
Test	*Diseased*	True Positive b_0	False Positive a_0
	Healthy	False Negative b_1	True Negative a_1

Table 1.1 Single threshold contingency table

The test can then be characterised by two ratios:

$$Hit\ Rate = \frac{True\ positive}{True\ positive + False\ negative} = \frac{b_0}{b_0 + b_1}$$

$$False\ Alarm\ Rate = \frac{False\ positive}{False\ positive + True\ negative} = \frac{a_0}{a_0 + a_1}$$

The *Hit Rate* is the fraction of the total diseased cases the system gets right. The *False Alarm Rate* is the fraction of the total healthy cases the system gets wrong, i.e. misclassifies as diseased. In medical nomenclature the Hit Rate is refered to as the *sensitivity*, and 1 – False Alarm Rate is refered to as the *specificity*. This thesis will use the terms 'Hit Rate' and 'False Alarm Rate'.

If multiple thresholds are used, for example, to categorise events into '*definitely diseased*' '*possibly diseased*', '*possibly healthy*', and '*definitely healthy*' the contingency table can be expanded to a 2 by n+1 table, where n is the number of thresholds; for example, to a 2 by 4 table, as shown in Table 1.2, where ϕ is the measurement.

		Standard	
		Diseased	Healthy
Test	$-\infty < \phi \leq$ Threshold 0	b_0	a_0
	Threshold 0 $< \phi \leq$ Threshold 1	b_1	a_1
	Threshold 1 $< \phi \leq$ Threshold 2	b_2	a_2
	Threshold 2 $< \phi \leq +\infty$	b_3	a_3

Table 1.2 Three threshold contingency table

From the table, n pairs of Hit Rate and False Alarm Rate can be calculated. For example, Table 1.2 gives the following three pairs:

$$Hit\ Rate_0 = \frac{b_0}{b_0 + b_1 + b_2 + b_3} \qquad False\ Alarm\ Rate_0 = \frac{a_0}{a_0 + a_1 + a_2 + a_3}$$

$$Hit\ Rate_1 = \frac{b_0 + b_1}{b_0 + b_1 + b_2 + b_3} \qquad False\ Alarm\ Rate_1 = \frac{a_0 + a_1}{a_0 + a_1 + a_2 + a_3}$$

$$Hit\ Rate_2 = \frac{b_0 + b_1 + b_2}{b_0 + b_1 + b_2 + b_3} \qquad False\ Alarm\ Rate_2 = \frac{a_0 + a_1 + a_2}{a_0 + a_1 + a_2 + a_3}$$

With a sufficiently large number of thresholds changing in small discrete steps, as might be obtained from a neural network, a plot of Hit Rate (along the *y* axis) against False Alarm Rate (along the *x* axis) for each threshold gives a receiver operating characteristic (ROC) curve. A typically shaped curve for a multi threshold plot is given in Figure 1.2.

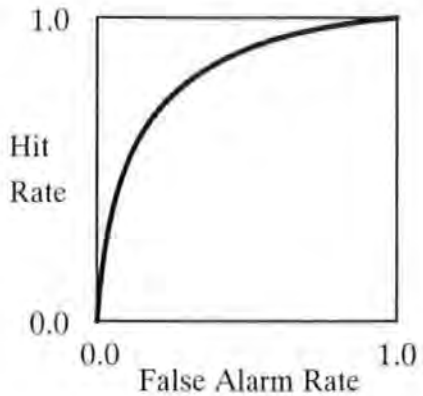


Figure 1.2 Example ROC curve

The curve thus shows the trade–off between correctly detecting diseased cases, and mistaking healthy cases for diseased cases. If the two underlying population distributions, as measured by a system, are completely separated, the curve will immediately rise to the top left corner (0.0, 1.0), and then proceed horizontally (Figure 1.3 (a)). If the distributions tend to overlap, so that healthy and diseased cases cannot be distinguished by the measurement, the curve will approach the diagonal (0.0, 0.0 to 1.0, 1.0) (Figure 1.3 (b)).

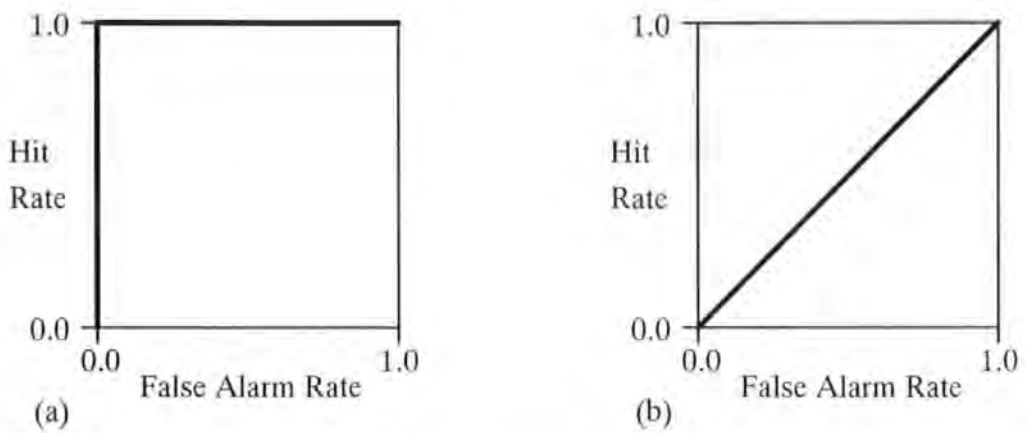


Figure 1.3 ROC curves of well separated and identical populations

1.1.1 Parametric and Nonparametric Curves

If there are only two test categories, as illustrated in Table 1.1, there is only one data point to plot as shown in Figure 1.4 (a). The four categories in Table 1.2 will generate a plot with three points as shown in Figure 1.4 (b). If a curve is to be generated from a limited number of points, then assumptions must be made about the form of the curve – either a smooth curve must be fitted (Figure 1.4 (b)) or straight line segments must be used (Figure 1.4 (a)).

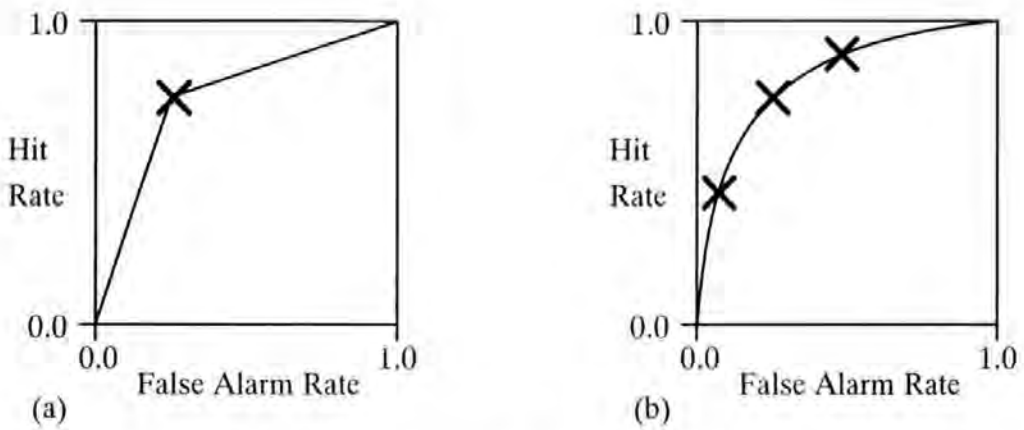


Figure 1.4 Nonparametric and parametric ROC curves

In order to fit a smooth curve through a limited number of points assumptions must be made about the parameters of the curve. In ROC analysis these parametric assumptions are actually made about the distribution of the diseased and healthy populations, which are then used to generate the parameters of the curve. A common assumption is that the distribution of the underlying populations are Gaussian (normal), which gives a '*binormal*' ROC curve. In the binormal model the shape of the curve is then fully characterised by two parameters:

- The difference between the mean of the healthy and diseased populations;
- The ratio of the standard deviation of the healthy and diseased populations.

A nonparametric curve makes no assumptions about the underlying distributions and the curve is plotted as straight line segments.

All ROC curves, whether parametric or nonparametric, start from an implicit first point at the coordinate (0.0, 0.0), and end at an implicit last point at the coordinate (1.0, 1.0).

The implicit first point has a threshold measurement of minus infinity (see Figure 1.1). This is below either distribution, so the point is at coordinate 0.0, 0.0 on the ROC graph. The implicit last point has a threshold measurement of plus infinity, which is above both distributions, so the point is at coordinate 1.0, 1.0 on the ROC graph.

1.1.2 The Area Under the Curve (AUC)

The Area Under the Curve (AUC) is often used to give a summary statistic of a ROC curve. The meaning of this summary statistic was explained by Green and Swets [89], as being the probability of ranking the case correctly when presented with a pair of cases, one known to be diseased and the other known to be healthy. This definition can be extended. It is equal to the average probability of identifying the diseased case when presented with every possible pair of one diseased case and one healthy case from an equal number of diseased and healthy cases. If there are equal numbers of diseased and healthy cases in the population the frequency of disease in the population can be said to be 50%. The AUC is thus a measure of overall diagnostic ranking normalized to a frequency of disease of 50%.

Conversely, it is observed that exactly the same ROC curve is obtained (given the same underlying healthy and diseased distributions (see Figure 1.1)) no matter what the frequency of disease. Thus both the ROC curve, and its AUC, are independent of the frequency of disease in the population. Hence the AUC is a good way to summarise the overall accuracy of a system, but this may not always be what is required.

1.1.3 Clinical Importance of ROC Points

While the AUC gives the overall accuracy of a test (normalized for 50%/50% diseased/ healthy), clinically it may be more important to consider particular points on the curve [16][17], or particular regions of the curve [18].

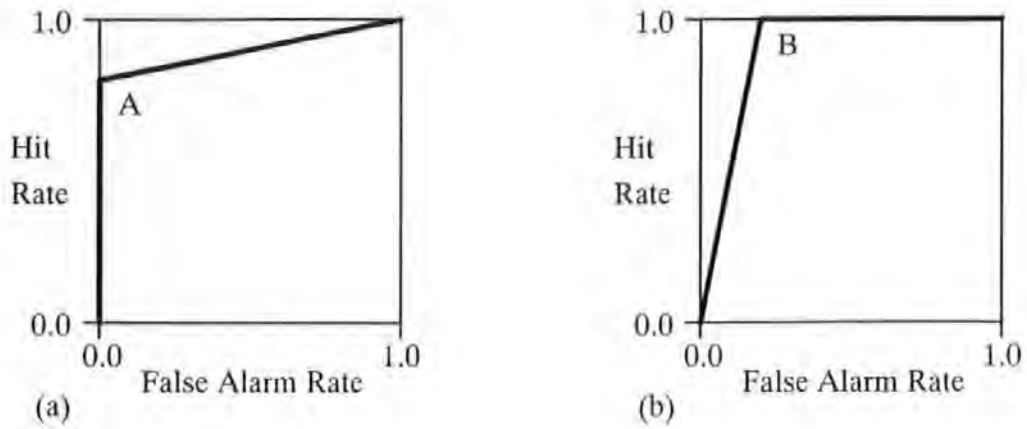


Figure 1.5 Clinically significant ROC points

For example, both curves (a) and (b) in Figure 1.5 have an AUC of 0.9, but radically different clinical utility. If a patient tests positive to a test characterised by ROC curve *a*, at point A, he/she has a 100% chance of having the disease, and a 0% chance of not having it. This can be a reassurance when taking the decision to use a treatment with considerable side effects on a serious disease. On the other hand, if a patient tests positive to a test characterised by ROC curve *b*, at point B, he/she has an 83.3% chance of having the disease, and therefore a 16.6% chance of not having it, which is unsettling in the situation described.

The figures above assumes the prevalence of disease in the population is 50% (as per the normalizing interpretation of the AUC). If the prevalence of disease is actually only a third of this, the chance (test *b*, point B) of disease being present is only 50%.

It is thus as important in ROC analysis to provide accurate information about points on a curve as it is to provide accurate information about the AUC.

1.1.4 Nonparametric and Parametric AUC Differs

The Area Under the Curve may vary depending on whether parametric or nonparametric assumptions are made. If the AUC is greater than 0.5, the parametric curve is convex and encloses a larger area than the straight line segments of the nonparametric curve (Figure 1.6). If the AUC is less than 0.5, the parametric curve is concave and the nonparametric AUC will be larger.

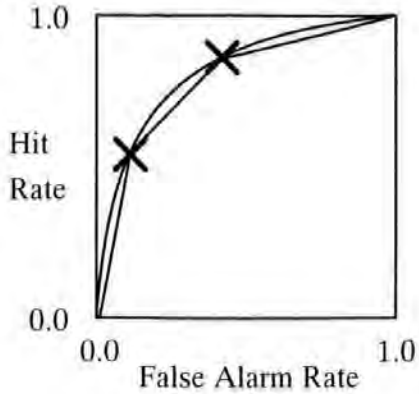


Figure 1.6 The parametric AUC is greater than the nonparametric AUC

Further introductory material on ROC curves can be found in van Erkel and Pattynama [16].

1.2 Confusion Matrices

1.2.1 Unweighted Confusion Matrices

In the more complex case of differential diagnosis between several diseases there are more than two categories, e.g. the categories might be '*Disease A*', '*Disease B*', or '*No disease*'. A contingency table, as shown in Figure 1.7, can be used to show the 'confusions' or mistakes, that can occur.

		Standard		
		A	B	None
Test	*A*	a_0	a_3	a_7
	B	a_5	a_1	a_4
	None	a_8	a_6	a_2

Figure 1.7 Example confusion matrix with labelled cells

The confusion matrix of a good discriminating system should show a high proportion of cases across the diagonal, i.e. the dark grey cells a_0, a_1, a_2 of Figure 1.7, and very few cases in the rest of the matrix (the white cells a_3 to a_8).

1.2.2 Weighted Confusion Matrices

Unweighted confusion matrices make the assumption that all misclassifications have equal consequences, but suppose that the drug to treat disease A is mildly beneficial to disease B. This is reflected in Figure 1.8 where cell a_3 is highlighted in grey. If the drug to treat B is cheap, and has no side effects in a healthy person, this misclassification may be of similar consequences, and hence cell a_4 is also shaded grey. The other misclassifications (giving the drug for A to a healthy person (a_4), of failing to diagnose disease A or B (a_6, a_8), or giving the drug for B to a person with disease A (a_5)) have equivalent consequences and are all shown in white in Figure 1.8.

		Standard		
		A	B	None
Test	*A*	a_0	a_3	a_7
	B	a_5	a_1	a_4
	None	a_8	a_6	a_2

Figure 1.8 Example weighted confusion matrix with labelled cells

These issues of clinical consequence and their financial implications are an area fraught with even more difficulty than determining standard diagnoses for test cases. Chapter 2 takes a closer look at this issue.

As far as evaluating a system is concerned, only the result of these deliberations over clinical consequences are required in terms of a numeric weight for each cell. If there are only two outcomes – '*right*' or '*wrong*', the weights 0 and 1 are implied, and the confusion matrix can be analysed using Cohen's kappa statistic [13] to give a measurement of agreement. Cohen's weighted kappa statistic [14] is available if more than two weights are specified.

1.3 Bayesian Posterior Intervals

One of the purposes of testing an intelligent medical system on a sample of test cases is to predict the performance of the system on the general population of patients if the system were to be used for real.

If an intelligent medical system was tested on 100 cases known to be diseased and gave 10 false negatives and 90 true positives, common sense would suggest that if unleashed on the general population of 1,000 diseased patients it is likely to pick up 900 and miss 100. However, there might have been an unlucky mix of test cases where there were more difficult cases (those with a low value of whatever is being measured, despite being diseased) than in the general population, or visa versa. What needs to be determined is the probability of obtaining a given accuracy, when in use in the population, given the uncertainty of the sample. As will be explained in Chapter 3, it can be calculated that there is a 95% chance that the system will incorrectly diagnose between 51 and 169 diseased patients as healthy (given the figures above), in other words, the 95% Bayesian posterior interval is 51–169, or 5.1%–16.9%. In order to deploy the system, a decision has to be made that a bet of 95% of misdiagnosing between 5.1% and 16.9% of the healthy population is worth taking.

All the measures above – the AUC of a nonparametric ROC curve, the AUC of a parametric curve, kappa and weighted kappa – have well established methods for calculating the 95% Frequentist confidence interval (which is commonly thought to be the same as the Bayesian posterior interval, but is actually based on a different paradigm, as will be explained in section 3.2.1). However, the accuracy of all these frequentist methods asymptotically increases as the sample size increases. At an infinite sample size they are all accurate, but at lower sample sizes the confidence interval is only an approximation. With the statistics above, it is very likely that the approximation is worse the higher the accuracy of the intelligent system measured, ie. the higher the AUC of a ROC curve, or the kappa statistic of a confusion matrix. A larger sample size is thus needed to compensate.

Obuchowski and Lieber [19] used Monte Carlo simulation to test the 95% Frequentist confidence interval of the AUC of ROC curves using up to 70 diseased and 70 healthy

cases. Some of their results are reproduced in section 3.6.1. Their work clearly demonstrates lack of accuracy at low sample sizes. Altaye et al. [20] used Monte Carlo simulations to test the 95% Frequentist confidence interval of the kappa statistic and while they generally obtained good results, they noted a small systematic error with changing values of kappa.

One of the problems with evaluating intelligent medical systems is the difficulty of obtaining the clinical opinion of rare and busy medical experts on a sufficiently large sample of cases to do a valid test. The fact that the existing statistics perform poorly with small samples only exacerbates the problem.

1.4 Aims and Objectives

1.4.1 Aims

The aim of this research was therefore to develop novel, robust, statistical methods for the evaluation of intelligent medical systems particularly where high accuracy systems are evaluated with small sample sizes.

1.4.2 Objectives

Specifically, the objectives were to research novel methods for calculating exact Bayesian posterior intervals, valid for any sample size, for the following five statistics that can be used in the evaluation of intelligent medical systems:

- The points on a nonparametric ROC curve;
- The comparison of corresponding points on two uncorrelated nonparametric ROC curves;
- The nonparametric AUC of a ROC curve;
- The parametric AUC of a ROC curve, and the two parameters of the curve;
- The weight of a weighted confidence matrix.

1.5 Overview of the Thesis

This research focuses sharply on particular aspects within the vast subject of intelligent system evaluation. Chapter 2 will place this research in context. It will define what is meant

by an intelligent medical system, explain some of the ethical, legal and psychological reasons why thorough evaluation is necessary, and discusses the difficulty of obtaining large numbers of test cases. The particular difficulties of determining a standard benchmark diagnosis for test cases will then be discussed, and finally, the even more difficult subject of assigning costs and benefits to medical choices that might provide the weights in a confusion matrix is examined.

Chapter 3 will take a more detailed look at ROC analysis and explain the most common existing statistical methods. The chapter starts with a short introduction to basic probability theory that will be used to explain existing ROC statistics and as the basic theory underpinning the novel methods presented in Chapter 5. Chapter 3 then explains the calculation of the Bayesian posterior intervals for each point of a ROC curve, which will be used as the starting point for the novel research explained in Chapter 5. The difference between a Bayesian posterior interval and a Frequentist confidence interval is then explained, before moving on to the basic theory behind the calculation of the existing Frequentist confidence interval of the nonparametric AUC. Methods for estimation of the parametric parameters and the AUC will then be examined. Finally the testing of these methods using Monte Carlo simulations is discussed.

Chapter 4 examines the statistics of confusion matrices. It describes the kappa statistics as a measure of agreement and how they are an improvement of previous statistics which only measured association. Monte Carlo simulations for testing kappa statistics are then explained.

Chapter 5 introduces the new methods of calculating Bayesian posterior intervals. The first section presents a more formal analysis of the posterior intervals of one point on a ROC curve than was given in Chapter 3, but adds a significant improvement. The next section presents novel work on the extension of the method to each point on a multi-point ROC curve. This result is then used to generate a method of producing the posterior interval of the difference between pairs of points on two different uncorrelated ROC curves. The insight gained into the mathematical structure of the problem is then used to devise an algorithm for generating the posterior interval of the nonparametric AUC. This algorithm

is generic and it is then adapted to calculate the posterior interval of the parametric AUC and the posterior interval of the paramteric parameters. Finally, the confusion matrix is examined. An unweighted confusion matrix has the same mathematical form as individual points on a nonparametric ROC curve, but the weighted confusion matrix requires another adaptation of the generic algorithm used to generate the nonparametric AUC.

Chapter 6 documents the Monte Carlo simulations used to validate the novel algorithms explained in Chapter 5 and presents some of the results. The full results are tabulated in the Appendix B because of their length.

Chapter 7 demonstrates applications of each of the novel methods explained in Chapter 5. Firstly, two ROC curves found in the literature, one with a small sample size from Adlassnig and Scheithauer [10], and the other with a large sample size from Swets [21], are compared and contrasted. Secondly, the weighted confusion matrix algorithm is demonstrated using data originally used by Cohen [14] as an example of weighted kappa.

Chapter 8 presents a brief summary of the novel methods and their utility. Section 8.3 discusses future work, particularly two analytic solutions to computational methods presented in Chapter 5. The first is a potential solution to calculating the posterior interval of the nonparametric AUC, the second a potential solution to the posterior interval of a weighted confusion matrix. Finally, the potential impact both for intelligent medical system evaluation and in the wider context is discussed.

2 Evaluation Issues

In order to test an intelligent medical system using a ROC curve, or a confusion matrix, a suite of test cases must be prepared with model answers, and any weights in a confusion matrix must be assigned. The cases can then be presented to the system and its responses recorded. It is also desirable for experts to give their opinion on the test cases in order to compare expert and system performance. This whole process is particularly difficult with intelligent medical systems. This chapter explains why.

Firstly, an explanation of what is meant by an 'intelligent medical system' is given. The medical role of these systems means that they are safety critical and must be tested to a high standard. The value of such systems is their 'intelligence' (or 'expertise'), but this limits the number of people who are qualified to judge the system, or to be compared against it, to a handful of experts. Thus the core problem is the difficulty of obtaining enough test data, and comparisons, to prove the system is safe and effective.

Secondly, even given willing experts with time to spare, preparing test cases, and assigning clinically meaningful measures to medical decisions (and errors), is not easy. Medical knowledge is largely opinion, where even experts can disagree, and assigning values to decisions (confusion matrix weights) is clinically, financially and morally difficult.

2.1 Intelligent Medical Systems

For the purposes of this thesis, both intelligent medical systems and medical expert systems will be loosely defined as any computer based tool, designed to help medical staff in their duties, that function by using artificial intelligence techniques to emulate human expertise. This is in contrast to medical information systems, e.g. medical record systems, that automate record keeping and other tasks without any special intelligence.

Artificial intelligence techniques can be potentially applied to a wide variety of tasks in medicine, from interactive diagnostic and prescriptive expert systems like MYCIN [22], to diagnostic machine vision systems for mammography [23] or cervical smear screening programs [24], to intelligent intensive care monitors [25]. The evaluation of each, for

safety and fiscal cost, requires the same set of factors to be examined, but with different emphasis.

2.1.1 Some Examples of Intelligent Medical Systems

An interactive diagnostic system, that may be used in a hospital by junior doctors when their senior colleagues are unavailable, needs to be user friendly and quick to use, while giving a sufficiently better diagnosis than the junior doctor could manage on his/her own, to be worth using. The advice given has to make sufficient difference to patient care that the hospital would prefer to install the intelligent medical system in every relevant ward, rather than employ another senior clinician, or bear the cost of hesitant junior doctors ordering more diagnostic tests than strictly necessary. The safety of the system has to be ensured so that it never misleads a junior doctor into giving inappropriate and dangerous treatment to a patient.

In a busy laboratory screening 1000 mammograms a week, an intelligent machine vision system could be used to increase through–put to 5000 per week even if it was not as good as the human experts. Provided it could filter out 80% 'barn door normals', with very high reliability the 20% of cases it was unsure about could be handled by the human experts as before.

Attaching intelligent alarms to monitoring equipment is another area where intelligent systems can have obvious benefit. Here each patient effectively gets a dedicated expert watching their monitor 24 hours a day, rather than a human expert glancing at it occasionally. Here the 'expertise' must be similar to diagnostic systems, in that all dangerous situations must be reported, but without so many false alarms that the physicians are forced to spend more time examining the intelligent monitors than they could afford to spend interpreting their dumb predecessors.

The logical extension of an intelligent monitor is one that then takes direct corrective action e.g. infusing drugs, changing the frequency of a ventilator, changing the flow of oxygen etc. In this case the safety factor cannot be over emphasised. The system can never be allowed to perform worse than the human expert it replaces.

All of these systems have obvious safety implications, from misleading diagnosis to dangerous treatment. All such intelligent medical systems are therefore subjected to ethical and legal considerations about their safety.

2.1.2 Safety and the Ethical Basis of Medicine

According to Wyatt and Spiegelhalter [26], the ethical basis of medicine is to strive to improve patients' health while attempting to 'do no harm', and to use limited health care resources wisely.

Wyatt and Spiegelhalter [26] also raised the issue of legal liability. At that date (1990) it was unclear if an intelligent medical system would be classified as a product or as a service. If the former, strict product liability laws dictate it must be safe, though a 'state of the art' defence might provide some mitigation; if the latter, then it must reach the standard of 'informed and sensible body of opinion'.

Finding a standard for an 'informed and sensible body of opinion' is one of the main issues in the field. As will be shown later, expert physicians do not always agree. As a consequence this thesis will take the pessimistic view that medical science is a body of opinion, not of fact. The problem is therefore, to both define the 'informed and sensible body of opinion' and then to make sure the intelligent medical system corresponds to it. In addition, given the ominous shadow of the unforgiving product liability laws, the closer the 'informed and sensible body of opinion' can be shown to be to the unobtainable absolute truth, the better.

Our own common sense only reinforces the point. None of us can relish the prospect of being treated by a doctor acting on misinformation from an intelligent medical system.

2.1.3 Safety Net

There appears to be a widespread opinion that physicians and other medical personnel are the final safety check on erroneous intelligent systems and that physicians both expect, and will take, the final responsibility for medical decisions [27][28].

Whitbeck and Brooks [28] make a distinction between 'descriptive' and 'prescriptive' expert systems, with a distinct preference for 'descriptive' systems that explicitly leave the decision with the doctor.

Psychological tests indicate that this is a moot point. Murphy and Yetmar [29], tested the faith accountants would place in financial reports prepared by juniors supported by, or not supported by, expert systems. The experiment consisted of obtaining opinions on four categories of reports:

- Known to the experimenters to be correct; known to the subjects as having been prepared with the help of an expert system;
- Known to the experimenters to be erroneous; known to the subjects as having been prepared with the help of an expert system;
- Known to the experimenters to be correct; known to the subjects as having been prepared without an expert system;
- Known to the experimenters to be erroneous; known to the subjects as having been prepared without an expert system.

The subjects were found to place more faith in reports prepared with the expert system, whether right or wrong.

Timnick [30] gave subjects arithmetic calculations, and then let them check their work with incorrectly programmed calculators. Significant numbers changed their own correct answers to the incorrect ones provided by the calculator. The example is remote from intelligent systems but still makes the point that in certain circumstances humans have a tendency to believe machines in preference to their own conclusions.

In light of this psychological evidence that humans are perhaps more likely than first assumed to be misled by an erroneous intelligent medical system, the importance of making sure the system output always conforms to an 'informed and sensible body of opinion' is only reinforced.

2.1.4 The Difference from 'Conventional' software

If intelligent medical systems are to conform to an 'informed and sensible body of opinion' they need to be tested against the existing 'informed and sensible body of opinion', on a large number of cases, and be seen to give the comparable answers.

In an ideal world, hundreds of thousands, to millions of cases should be put through intelligent systems, to bring them up to 'product liability' standard, as is common with non-intelligent software systems, but the unique property of these systems, their 'intelligence', means they can only be properly compared with human experts, who are both rare and busy. This is in complete contrast to the testing of other software systems, say stock control systems, or navigation systems, in that the basic logic of simpler systems can be understood by software engineers and testers to the extent that large test sets can be produced manually or semi-automatically, unaided by experts. It is ironic that the more 'intelligent' the system, the smaller the number of experts available who are capable of producing the test cases necessary to prove the system is actually 'intelligent'.

It is an unfortunate fact of medical life that the top experts required to provide an informed opinion are often too busy to do so. For example, in the second trial of the MYCIN system [22], some of the evaluators, who were asked to give their opinion on only 15 cases, and were paid an honorarium to encourage them, took a year to return the evaluation forms. This factor is a real, and serious, constraint on evaluating intelligent medical systems.

An alternative strategy is to validate against existing databases of patient records. The problem then shifts to that of collecting an accurate and extensive database from busy clinicians who have little reason or incentive to add data logging tasks to their already heavy work load. It should also be noted that junior staff will rely on an expert system far more for rare and difficult cases than for routine cases with which they are familiar. The database therefore has to be large enough to contain a statistically valid sample of these rare, but important cases, which usually means logging a huge number of routine cases as well.

2.2 Quantifying Medical Care

In order to properly quantify the change to clinical practice brought about by the introduction of an intelligent medical system, the effectiveness of clinical practice, with particular regard to the safety of patients, and the cost of the clinical practice, must be measured. A useful intelligent medical system will increase clinical effectiveness, while reducing cost. While we have a measure of financial cost, e.g. pounds Sterling, it is often difficult to apportion costs. There is even greater uncertainly in finding practical measures of clinical effectiveness.

2.2.1 Fiscal Cost

Of the two parameters, cost and clinical effectiveness, cost is both possibly the easier to measure and the most contentious. The issue raises the hope of a Utopian dream of 'unlimited healthcare for all', however the pragmatics of economics are less kind.

Maynard [31] has observed that for every 1% rise in gross domestic product in the U.K., health care spending rises by 1.2%. Richer countries tend to spend more in absolute terms as well as in relative terms. As a percentage of gross domestic product the USA spends 15%, the French 8.8%, the Germans 8% the Japanese 6.5% and the UK 6.2%. Even the comparatively modest spending in the UK has risen by nearly 22% over the fifteen years between 1978 and 1993 [31]. It is inevitable that continuous demand for a scarce resource like healthcare will be regulated by capitalist economics or social engineering. This places pressure on intelligent medical system evaluation to demonstrate the system will be economical in use.

2.2.2 Clinical Effectiveness

Clinical effectiveness is emotively regarded as far more important than pure cost considerations but is very hard to measure.

The most straight forward method of objective assessment is to measure the effect of a treatment on one of the symptomatic clinical parameters of a medical condition or disease, for instance, reduction in blood pressure for hypertension drugs. Similar localised effects may be measurable for intelligent systems. However, such effectiveness criteria do not

give a clear indication of the actual effect on the patient's life, after all, a person's blood pressure is of no concern to them apart from when it causes inconvenience, pain, or death. If a patient has a high blood pressure 150% of normal, and a cheap drug reduces pressure to 105% of normal, while an expensive drug reduces pressure to 103% of normal, which drug is best? For that a measurement of the final outcome, the effect on life or death, should be used. One of the most common measures of final outcome, especially of serious medical conditions, is survival rate. This is often 5 year survival, but can be any period of time.

Outcome criteria such as survival rate do not tell the full story, quality of life is important as well. There may be little point in prolonging a terminally ill patient's life for a few weeks if they will have very low quality of life during the period. The Quality Adjusted Life Year (QALY) [32] is a measure of quality of life over time. For example, if a treatment for arthritis will improve a individual's health from 70% to 90% of perfect health for a year, the utility of the treatment is 0.2 QALY.

There are also problems in measuring quality of life, as it depends on the individual. How much is a clerical worker disabled by a stiff index finger as compared to a piano player with the same problem? There are also serious ethical difficulties. For instance how ethical is it for person A to value person B's life?

Another proposed measure is the micromort [33]. One micromort is a chance of one in a million of sudden painless death. It can be used to generate comparative utility values by asking questions of the form:

'*If we use the old surgical technique you will be in hospital for 10 days, if we use the new technique you will be in hospital for 3 days, but there are x chances in a million that you will die on the operating table, what does x have to be for you to have no preference between the alternatives*?'

'*Drugs A and B are identical in effect, except drug A causes nausea, while drug B has x chances in a million of causing sudden unconsciousness and death, what does x have to be for the alternatives to be equally attractive?*'

These two examples give the patient's opinion of the negative utility of staying in hospital, and the patient's opinion of the negative utility of nausea as a drug side effect, expressed

in a common unit. The problem is that every individual will have different priorities, but some idea of average relative 'utility' for the patient population as a whole could be constructed.

In practice, the difficulty of these measures of overall clinical effectiveness are so great that they have not been used to take clinical decisions of treatments or clinical systems in the U.K. However, there is an increasing realisation that some type of overall objective measurements of clinical effectiveness is necessary to gain maximum benefit from limited resources.

2.3 Obtaining Consensus

2.3.1 The Need for Expert Opinion

In order to make sure an intelligent medical system conforms to an 'informed and sensible body of opinion' that opinion has to be gathered. It is obvious that some kind of testable, factual basis should be sought in the first instance, and in some cases such a 'Gold Standard' can be found, but sometimes medical knowledge is just opinion, and so somehow a consensus opinion must be obtained. The 'Delphi Protocol' is a method for obtaining such a consensus opinion [34].

2.3.2 Gold Standards

The ultimate theoretical measure of an expert system is not how close it comes to existing expert opinion, but how close it comes to absolute truth. The standard of absolute truth is usually called the 'Gold Standard'. In the absence of a 'Gold Standard', truth has to be approximated by a 'Silver Standard'. Phelps [35] used the term 'Fuzzy Gold Standard' to emphasise the inherent uncertainty in either a 'Gold Standard' or 'Silver Standard' against the unobtainable absolute truth.

Swets [21] illustrates the difficulties with 'Gold Standards' using examples from medical imaging. The 'Gold Standard' for malignant lesions is obtained by tissue examination of a biopsy obtained at surgery or autopsy. The 'Gold Standard' for benign lesions is long term follow up. Difficulties arise in that the radiological diagnosis is not correlated in either space or time with the pathology. Different tissue might be sampled, and it will certainly

be taken at a different time. A pathological abnormality may not have been present when the image was taken, or an abnormality on an image may have regressed. The pathologists code or language for describing lesions may not map precisely onto that used by radiologists. Subjects can drop–out from long term follow up, or might develop malignancy subsequent to a 'Gold Standard' benign image being taken.

The only accurate 'Gold Standard' is where a test is the medical definition of a disease. This is only possible if all experts can agree on that definition. All others are approximations, to a greater or lesser extent, of the absolute truth and often have drawbacks, such as being risky, or painful to the patient, or being expensive, or taking a long time.

2.3.3 Delphi Protocol

Where objective tests for a standard do not exist, expert opinion is the only alternative. The problem then becomes one of finding a consensus expert opinion. In these circumstances the Delphi Protocol [34] can be used to create a standard for a group of test cases. In the literature this standard is usually referred to as a 'Silver Standard'.

The Delphi Protocol was developed to counteract the drawbacks of face–to–face group interaction, such as the influence of dominant individuals, reluctance to abandon a position once publicly taken, and group pressure towards conformity. It is characterised by anonymity and iteration with controlled feedback.

Firstly, each expert evaluates each case in isolation. If they all agree on a case, the consensus opinion becomes the consensus standard. The controversial cases left are anonymously marked with each expert's opinion, with reasoning, and resubmitted to all the experts. Again, if a case then achieves a consensus, the consensus becomes the standard. Any cases still remaining are discussed in an open forum of all the experts until a consensus is reached, which then becomes the standard. Thus all experts give their opinion twice before any psychological effects of personality can influence them.

2.4 Summary

This chapter has briefly examined the types of intelligent system used in medicine and highlighted some important factors:

- Intelligent medical systems need to be accurate – depending on their role, they must be shown to reach certain criteria for Hit Rate or False Alarm Rate, or both.
- Intelligent medical systems must be safe – at the very least, they must be shown to conform to an 'informed and sensible body of opinion'. To demonstrate this, systems must be compared to human experts and shown to give comparable performance.
- The more 'expert' a system, the fewer human experts there are capable of producing test cases.
- Human experts are often very busy, and only able to give their opinion on a limited number of test cases.
- Expert medical knowledge is rarely absolute, and it is often only opinion.

These comparisons must be made relative to an important consideration:

- It is important that the clinical effectiveness of the environment including the system is an improvement over the clinical effectiveness of the same clinical environment without the system. However desirable, meaningful measures of clinical effectiveness are difficult to obtain.

If systems are going to conform to an 'informed and sensible body of opinion' they need to be measured against the existing body of 'informed and sensible opinion' , which therefore needs to be determined.

- 'Gold Standards' are pragmatic approximations of 'absolute truth' based on pathology, long term follow up, or other studies.
- If there is only opinion, the 'Delphi Protocol' can be used to at least arrive at a consensus opinion, often called a 'Silver Standard'.

For the reasons discussed above, an intelligent medical system test is likely to be carried out with a small number of expertly considered test cases, from which the maximum

amount of information about system safety and effectiveness must be extracted. The rest of this thesis focuses on improved methods of extracting that information. In this respect the intelligent medical system can now be regarded as a 'black box' that takes a series of inputs and produced a set of outputs which are compared with the 'Gold Standard' answers.

Evaluation might not be done at the physical level of the hardware (e.g. a computer or piece of medical equipment with an embedded intelligent system), but might be done at a higher level. For instance, evaluation might compare the performance of a whole hospital ward plus intelligent systems, to the same type of ward without the system. However, this scenario can still be modelled as a 'black box' which takes admitted patients as input, and discharged patients as output, provided appropriate computer logged 'Gold/Silver Standard' categories can be assigned to the admitted and discharged patients.

3 ROC Statistics

This chapter documents the main existing methods of analysing ROC curves, demonstrates their weaknesses and introduces the very basic probability theory that will be used to develop powerful ideas in Chapter 5.

First basic probability theory is explained. This is used as both the basis for a rigorous understanding of existing ROC analysis, and as the basis of the novel methods explained in Chapter 5.

The three main methods of ROC analysis will then be explained:

- The Bayesian posterior interval of individual ROC points;
- The Frequentist confidence interval for the nonparametric AUC;
- The Frequentist confidence interval for the parametric AUC, and parameters.

During this section, the pivotal role of the Gaussian distribution, and its limitations with small sample sizes, will be discussed.

The chapter ends by examining how good these methods are in practice by looking at Monte Carlo simulations. Finally, it will be argued that these Monte Carlo tests are themselves limited.

3.1 Basic Probability Theory

Probability is an intuitively simple concept that for all its simplicity defies unambiguous definition. Probability can be explained as a proportion, a relative frequency, an expected value, or an opinion. In the current discussion it is only necessary to define probability as a proportion.

3.1.1 Probability as Proportion

An American Roulette wheel has 38 numbers, and (in the absence of tricksters) it would be expected that the ball has an equal chance of landing on each number. It can therefore be assume that the probability of a 7 is 1/38. Similarly, the probability that a flip of a fair coin gives a head is 1/2, or the probability of drawing one of the 4 aces from a pack of 52 playing cards is 1/13.

3.1.2 Probability Expressed as a Number between 0 & 1

The concept of probability can be expressed as a number in the range 0 and 1 inclusive. The probability of a 7 on a roulette wheel is thus $1/38 = 0.0263$.

3.1.3 Events

If a fair coin is flipped once there are two, and only two possible outcomes (if the possibility of the coin landing on its edge is ignored), a head or a tail, which can be represented by the sample space, Ω:

$\Omega = \{ H\ T \}$

Since these are the only possible events, one of them must occur. If the probability of a head is given by $P(H)$ and the probability of a tail by $P(T)$, then:

$P(H) + P(T) = 1$

$\therefore\ P(T) - 1 = P(H)$

If the coin is fair $P(H) = P(T) = 0.5$.

By definition, the sum of the probabilities of all outcomes in the sample space is 1.0.

3.1.4 Independence and the Product Rule

In a casino the chance of a 7 on the next spin of the roulette wheel is $1/38$, and the chance of the next card dealt being an ace is $1/13$. The events are totally independent, and the outcome of one cannot effect the other. The chance of both events happening is therefore the product of the two probabilities, which is $1/38 \times 1/13 = 0.00202$.

If a fair coin is flipped three times, the sample space can be represented by:

$\Omega = \{ HHH\ HHT\ HTH\ HTT\ THH\ THT\ TTH\ TTT \}$

and the probability of an event, say HHH, is given by:

$P(HHH) = P(H) \times P(H) \times P(H) = P(H)^3 = 0.125$

Similarly, the probability of the event HTH is thus:

$P(HTH) = P(H) \times P(T) \times P(H) = P(H)^2 \times P(T) = 0.125$

Now suppose the coin is not fair, but there is a 0.60 chance of a head, (and, by implication, a $1 - 0.6 = 0.4$ chance of a tail):

$$P(HTH) = P(H) \times P(T) \times P(H) = P(H)^2 \times P(T) = 0.6^2 \times 0.4 = 0.144$$

In general, if the probability of a head is x, the probability of a specific sequence of $a_0 + a_1$ flips in which there are a_0 heads, and a_1 tails, is given by:

$$x^{a_0} (1 - x)^{a_1}$$

However, there are many sequences with the same number of heads and tails. By inspection of the sample space of three flips it can be seen that there are three sequences with two heads and one tail:

$$P(HHT) = P(H) \times P(H) \times P(T) = P(H)^2 \times P(T) = 0.6^2 \times 0.4 = 0.144$$

$$P(HTH) = P(H) \times P(T) \times P(H) = P(H)^2 \times P(T) = 0.6^2 \times 0.4 = 0.144$$

$$P(THH) = P(T) \times P(H) \times P(H) = P(H)^2 \times P(T) = 0.6^2 \times 0.4 = 0.144$$

Only one of these events can occur, and therefore the probability of any one of them occurring, i.e. the probability of there being two heads and one tail irrespective of order, is the sum of the probabilities of these mutually exclusive sequences:

$$P(HHT) + P(HTH) + P(THH) = 0.144 + 0.144 + 0.144 = 0.144 \times 3 = 0.432$$

Determining the number of sequences with the same number of heads by inspection of the sample space is impractical for large numbers of flips. It needs to be calculated.

3.1.5 Permutations, Factorials and Binomial Coefficient

Suppose that five coins must be placed in a row. There are five choices for the first coin in the row. Once it is placed, there are four coins left and therefore four choices for the next coin in the row, giving a total of $5 \times 4 = 20$ ways of arranging the first two coins in the row of five. There are now three choices for the third coin, and this makes $5 \times 4 \times 3 = 60$ different ways of arranging the row so far. There are two choices of the fourth coin, and hence $5 \times 4 \times 3 \times 2 = 120$ arrangements of the first four coins. There is only one coin left to finish the row, thus the total number of arrangements of the row is $5 \times 4 \times 3 \times 2 \times 1 = 120$, which is called '5 factorial' and written '5!'. In general:

$$n! = \prod_{i=0}^{i=n-1} (n - i) \qquad \text{where} \qquad n > 0$$

Now suppose that two of the coins are 'identical'. If both these coins are removed, then replaced in the resultant gaps at random, the sequence will look identical to how it did before the coins were removed. There are two choices of coin to place in the first gap, and therefore there is one choice of which coin to place in the second gap. Thus there are only half (1/2!) the number of unique sequences than was first supposed.

Now suppose that the three remaining coins are 'identical' to each other. If all three coins are removed, and replaced at random, the sequence will still look the same however the coins are replaced. There are three choices to fill the first gap, then two choices to fill the second gap, etc. There are thus 3! ways of replacing the three coins, and thus only $1/3! = 1/6$ of the sequences previously thought to be unique actually are.

Now suppose the 'identity' of each coin is actually determined by whether it was a head or a tail on a random flip. Thus, if a coin is flipped five times, there are $5!/(2!\ 3!) = 10$ unique sequences with two heads and three tails. Inspection of the sample space of all 32 sequences that results from five flips will confirm this.

In general, if there are $a_0 + a_1$ events, each with two possible outcomes, head or tail, the number of sequences with a_0 heads and a_1 tails, is given by the binomial coefficient:

$$\begin{bmatrix} a_0 + a_1 \\ a_0 \end{bmatrix} = \frac{(a_0 + a_1)!}{a_0!\ a_1!}$$

3.1.6 Binomial Distribution

Combining the expression for the number of unique sequences of a_0 heads and a_1 tails, with the expression for the probability of a unique sequence, where the probability of a head is x, gives the probability of a_0 heads and a_1 tails irrespective of sequence:

$$\begin{bmatrix} a_0 + a_1 \\ a_0 \end{bmatrix} x^{a_0} (1 - x)^{a_1}$$

This is the result that is the foundation of this thesis. However, it will be found that it can be simplified further in the next section.

3.2 The Binomial and Nonparametric ROC

The basic theory above is directly applicable to ROC curves. Suppose there is a sample of patients known by a 'Gold Standard' to be disease free. A certain diagnostic process, or intelligent medical system, has an unknown probability, x, of incorrectly diagnosing each healthy case as diseased. The probability, p_x, of having a_0 patients mistakenly diagnosed as diseased and a_1 correctly diagnosed as healthy is therefore a function of x:

$$p_x = \begin{bmatrix} a_0 + a_1 \\ a_0 \end{bmatrix} x^{a_0} (1 - x)^{a_1} \tag{3.1}$$

A graph can be plotted of p_x against x. Figure 3.1 shows such a graph where p_x has been calculated for 1025 evenly spaced values of x (0, 1/1024, ..., 1023/1024, 1) for 8 true positives (a_0) and 16 false negatives (a_1). When viewed in this way, as a function of fixed values of a_0 and a_1, equation 3.1 (as plotted in Figure 3.1) is a *'likelihood function'*.

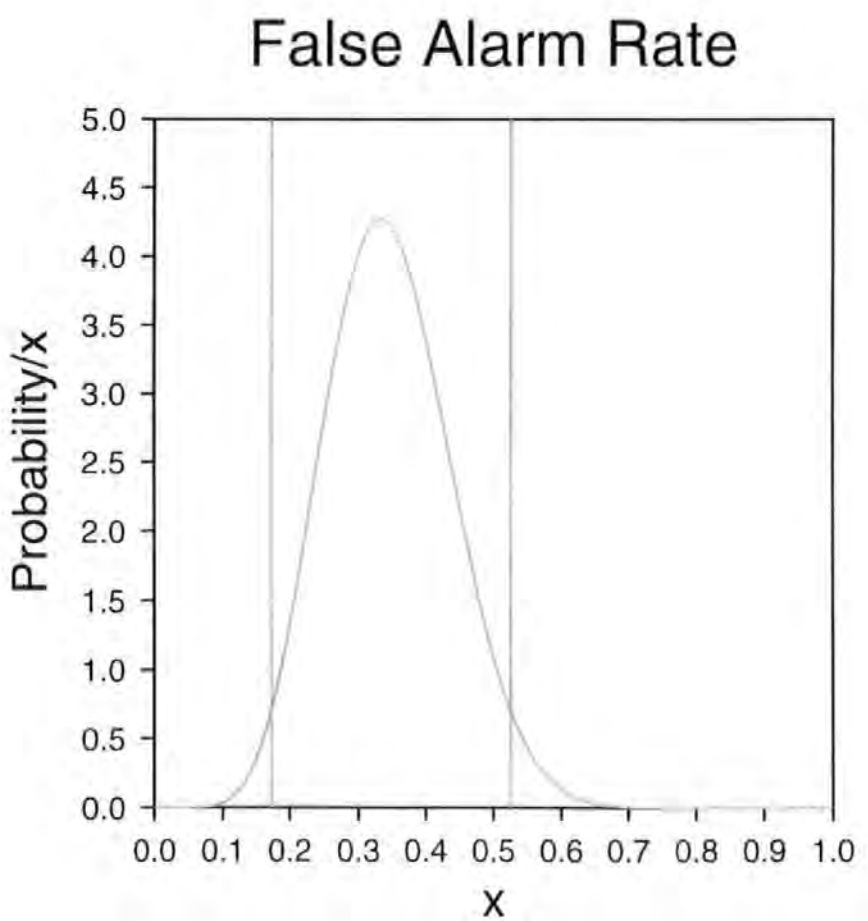


Figure 3.1 Likelihood function for 8 false positives and 16 true negatives

If the area under the graph is normalized to 1, this discrete plot will closely approximate the probability density function (pdf) of x. If the most likely value of x is picked, followed by the next most likely, and so on until the sum of the probabilities of the selected points equals 0.95, the range of x values chosen will give the 95% Bayesian posterior interval for x. The graph is actually already normalized, and the 95% Bayesian posterior intervals are shown by vertical lines at x values of 0.173 and 0.526. Thus, if 8 out of 24 healthy patients are misdiagnosed in an intelligent system test there is a 95% chance that the actual probability of the system misclassifying a healthy patient is between 17.3% and 52.6%.

A more precise graph can be produced by plotting the integral of the function, over a series of consecutive slices, which gives the pdf:

$$X_i = \frac{\int_{\frac{i-1}{n}}^{\frac{i}{n}} \begin{bmatrix} a_0 + a_1 \\ a_0 \end{bmatrix} x^{a_0} (1-x)^{a_1} \, dx}{\int_0^1 \begin{bmatrix} a_0 + a_1 \\ a_0 \end{bmatrix} x^{a_0} (1-x)^{a_1} \, dx} = \frac{\int_{\frac{i-1}{n}}^{\frac{i}{n}} x^{a_0} (1-x)^{a_1} \, dx}{\int_0^1 x^{a_0} (1-x)^{a_1} \, dx} \tag{3.2}$$

Where X is a vector of n elements.

Section 5.1 examines this in more detail and derives an expression for the integral. For now it should be noted that the binomial coefficients cancel out, so equation 3.2 can be simplified to:

$$p_x \propto x^{a_0} (1-x)^{a_1}$$

The denominator of equation 3.2 is the beta function, and the numerator is a partial beta function. The beta distribution has an orthogonal relationship to the binomial distribution. If p_x is plotted in two dimensions as a function of x and a_0 (where $a_0 + a_1 = n$), slices parallel to the x axis give a continuum of binomial distributions for different values of x, while slices parallel to the a_0 axis give $n + 1$ discrete beta distributions for values of $a_0 = 0, 1, ..., n$.

Considering the power and simplicity of the method it appears under utilised in ROC analysis given that it has been known for some time. The integral of the beta function was exam-

ined by Whittaker[36] at the turn of the century. In the 1930s, the use of the beta function for confidence limits was investigated by Fisher[37][38], and Neyman[39], and in the 1940s by Clopper and Pearson [40]. Murphy [41] gives examples of its use in manufacturing control. It would appear that only Hilgers [42] has used the beta function to calculate the Bayesian posterior intervals of the points on a ROC curve.

So far only the actual healthy cases have been discussed. Suppose the probability of an actual diseased case being correctly diagnosed as diseased is y. The probability of b_0 disease cases being diagnosed as diseased (true positives) and b_1 as healthy (false negatives) is therefore:

$$p_y \propto y^{b_0} (1 - y)^{b_1}$$

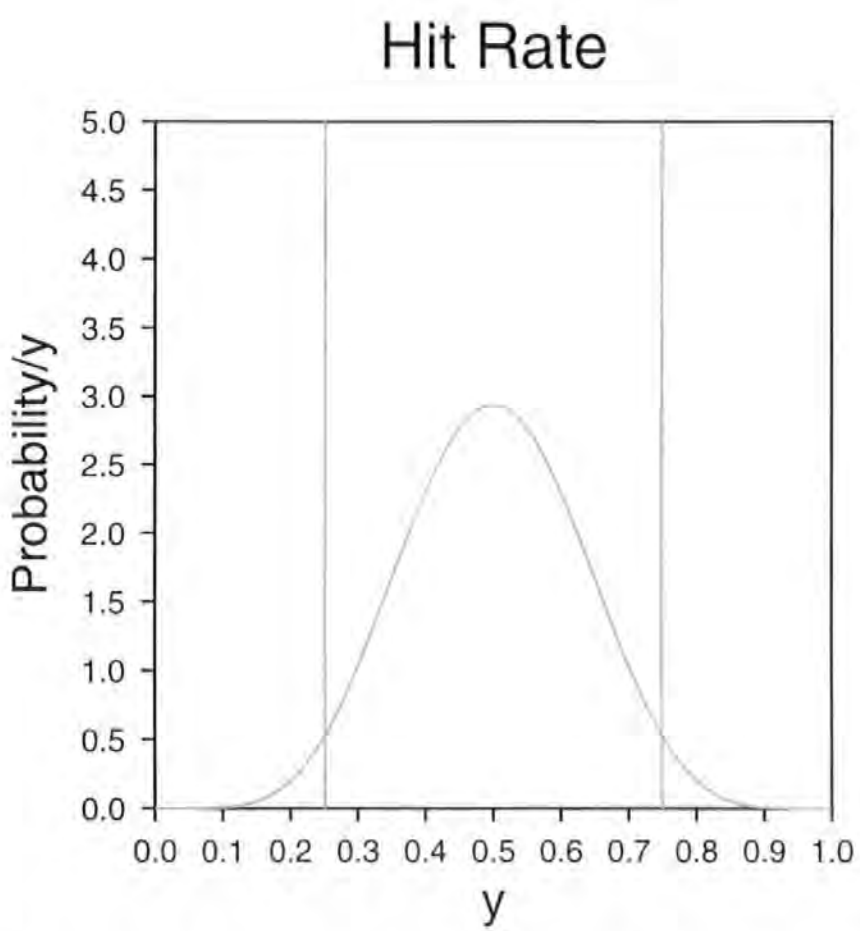


Figure 3.2 Likelihood function for 6 true positives and 6 false negatives

It is therefore obvious that exactly the same procedure can be applied to find the 95% Bayesian posterior interval of the population Hit Rate. Figure 3.2 shows the probability density

function for $b_0 = 6$ and $b_1 = 6$, which gives a 95% Bayesian posterior interval of 0.251 to 0.749.

3.2.1 Posterior Intervals and Confidence Intervals

The above approach is one paradigm for quantifying the degree of uncertainty of a statistical result. The Frequentist, or Classical, statistical paradigm looks at the problem differently [43].

It is assumed there is a unknown but fixed population False Alarm Rate that can be estimated to be 0.333 (given 8 false positives and 16 true negatives). Given this estimate of the population point, a Gedankenexperiment (thought experiment) can be performed where samples of 24 cases are repeatedly drawn from the healthy population with this False Alarm Rate. This will give a binomial distribution of samples with all possible combinations of false positives and true negatives for 24 cases, i.e. 0:24, 1:23, ... 24:0. If the most likely combination is picked, followed by the next most likely, and so on until the sum of the binomial probabilities of the selected combinations equals 0.95, the range of combinations will give the 95% confidence interval for the sample assuming a population False Alarm Rate rate of 0.333. In essence, it is assumed that the population point is fixed, and the sample is randomly drawn from the population, and therefore that the confidence interval shows where the sample will be 95% of the time if the experiment happened to be repeated a large number of times.

In contrast, the Bayesian paradigm discussed above regards the observed sample as fixed, and calculates the probability of generating the fixed sample for every possible population point. However, this paradigm carries an assumption which is examined in the next section.

3.2.2 Bayesian Prior

Suppose there is good evidence that some False Alarm Rates are impossible. For example, a system may be based on laboratory biochemical experiments that prove it is theoretically impossible to have a False Alarm Rate of 0.0. The model above has calculated the probability that each of 1025 False Alarm Rates (0, 1/1024, ..., 1023/1024, 1) gives the sample actually obtained, normalised to sum to 1.0. Now it is known that a False Alarm Rate of

0.0 cannot produce the sample at all. Therefore, the calculation changes to calculating the probability the 1024 False Alarm Rates (1/1024, 2/1024, ..., 1023/1024, 1) give the sample, letting a False Alarm Rate of 0 have a probability of 0.0 of giving the sample, and then normalising. This prior expectation can, in general, be any probability density function that models information or beliefs about the system before the experiment begins, and is called the *'Bayesian prior'*. The Bayesian prior can be accommodated by applying Baye's theorem to the logical process used above.

3.2.3 Baye's Theorem

Baye's theorem provides a way of calculating the probability of events conditional to their prior probabilities. Suppose that it is known that 30% of men die of a heart attack, 40% of men are overweight, and 80% of men who die of a heart attack are overweight. This can be expressed as:

$p(H) = 0.3$	Probability of a heart attack
$p(O) = 0.4$	Probability of being overweight
$p(O\|H) = 0.8$	Probability of being overweight conditional on a heart attack

Baye's theorem can now be used to give the probability of having a heart attack conditional on being overweight:

$$p(H|O) = \frac{p(O|H) \cdot p(H)}{p(O)} = \frac{0.8 \times 0.3}{0.4} = 0.6$$

Diagrammatically this can be seen in by Figure 3.3, where (as highlighted):

$$p(H|O) = 0.6 = \frac{0.24}{0.4} \qquad p(O|H) = 0.8 = \frac{0.24}{0.3}$$

		Overweight		
		Yes	No	Total
Heart Attack	Yes	0.24	0.06	0.3
	No	0.16	0.54	0.7
	Total	0.4	0.6	1.0

Figure 3.3 Diagrammatic Example of Baye's Rule

Baye's theorem can now be applied to calculating the probability density function in section 3.2. The probability that the population False Alarm Rate is x given the data a_0, a_1 is given by Baye's theorem as:

$$p(x|a_0, a_1) = \frac{p(a_0, a_1|x) \cdot p(x)}{p(a_0, a_1)} \tag{3.3}$$

Where the prior probability of a given value of x is given by (note that if $p(x)$ is a uniform distribution, $p(x)$ is constant for any value of x):

$p(x)$

The probability of obtaining the data a_0, a_1 given a value of x is therefore:

$$p(a_0, a_1|x) = \begin{bmatrix} a_0 + a_1 \\ a_0 \end{bmatrix} x^{a_0} (1 - x)^{a_1}$$

The probability of obtaining the data a_0, a_1, independent of x (by integrating out x) is:

$$p(a_0, a_1) = \int_0^1 p(a_0, a_1|x)\ p(x)\ dx$$

Therefore (from Baye's theorem (Equation 3.3)):

$$p(x|a_0, a_1) = \frac{p(a_0, a_1|x) \cdot p(x)}{\int_0^1 p(a_0, a_1|x) \cdot p(x)\ dx}$$

If, and only if $p(x)$ is a constant (i.e. a uniform distribution), the $p(x)$ terms cancel out:

$$\therefore\ p(x|a_0, a_1) = \frac{p(a_0, a_1|x)}{\int_0^1 p(a_0, a_1|x)\ dx}$$

Which is consistent with the explanation given in section 3.2 under the assumption of a uniform Bayesian prior distribution.

The uniform Bayesian prior will be used in the rest of this thesis, unless noted otherwise.

3.2.4 Definition of a Probability Density Function (pdf)

The discussion above has informally introduced the concept of a probability density function (pdf) which can be defined as a function that returns the probability, *Pr*, that a random variable is in a given range A to B:

$$\Pr = \int_{A}^{B} p(x)\ dx$$

A probability density function can be defined over an area, volume or hypervolume, as a function that returns a probability that a random variable is within a given area, volume or hypervolume respectively.

3.2.5 Projecting Posterior Intervals

Given a 95% posterior interval for the False Alarm Rate, and a 95% posterior interval for the Hit Rate it is possible to combine them to produce a posterior interval for one ROC point. Figure 3.4 shows a sketch of the rectangular posterior interval that might be derived from the data in Figures 3.1 and 3.2.

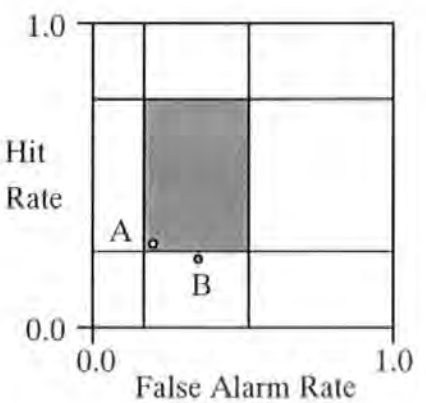


Figure 3.4 Rectangular confidence interval

Hilgers [42] used this method to plot a series of rectangular posterior intervals for each point on a ROC curve.

Rather than take this route, section 5.1 will examine an expression for the joint probability of the False Alarm Rate and Hit Rate. This expression can be used to calculate the two dimensional probability density function, and hence plot the two dimensional 95% posterior interval, directly from:

$$p_{xy} \propto x^{a_0} (1-x)^{a_1} y^{b_0} (1-y)^{b_1}$$

This assumes that the probabilities are independent. This will be examined in section 5.1.

3.2.6 Two Dimensional Posterior Intervals

If a posterior interval can be calculated for the Hit Rate of a ROC point, and another can be calculated for the False Alarm Rate of the same point it does not follow that the posterior interval in two dimensions has been completely specified.

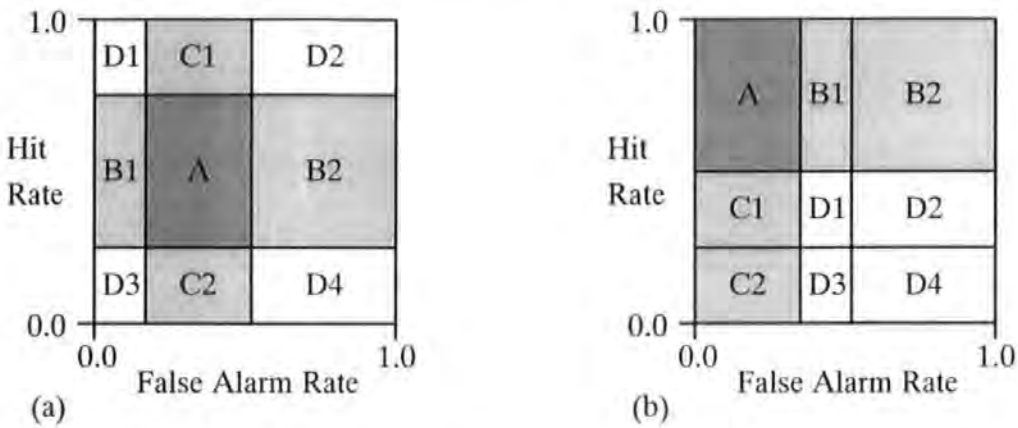


Figure 3.5 Analysing a rectangular confidence interval

Figure 3.5 (a) shows a ROC curve where the 95% posterior interval of the Hit Rate is given by the two horizontal lines bounding the region *B1*, *A*, *B2*, and the 95% posterior interval of the False Alarm Rate is given by the two vertical lines bounding the region *C1*, *A*, *C2*. Figure 3.5 (b) can now be obtained from (a). The left vertical region *D1*, *B1*, *D3* is swapped with the centre vertical region *C1*, *A*, *C2*, and the top horizontal region *D1*, *C1*, *D2*, is swapped with the centre horizontal region *B1*, *A*, *B2*. The (now adjacent) regions *B1*, *B2* can be combined to give region *B*; *C1* and *C2* can be combined to give *C*; and *D1*, *D2*, *D3* and *D4* can be combined to give *D*.

From the diagram:

$A + B = 0.95$

$A + C = 0.95$

$\therefore\ B = C$

From the diagram, it is also shown that:

$D + B = 0.05$

$D + C = 0.05$

Since $C = B \geq 0.0$ and $D \geq 0.0$

$\therefore\ 0.0 \leq C = B \leq 0.05$

If $C = B = 0.0$ then $D = 0.05$ and $A = 0.95$

If $C = B = 0.05$ then $D = 0.0$ and $A = 0.90$

$\therefore\ 0.90 \leq A \leq 0.95$

Thus the exact probability that a population point will occur in the region A cannot be given, it can only be given within bounds (Figure 3.6 sketches the two dimensional posterior intervals that could produce values of A of 90% and 95%). Both extremes are unlikely to occur in practice. Hilgers [42], used linear posterior intervals of 0.975, and hence his rectangular posterior intervals were in the range $0.95 \leq A \leq 0.975$.

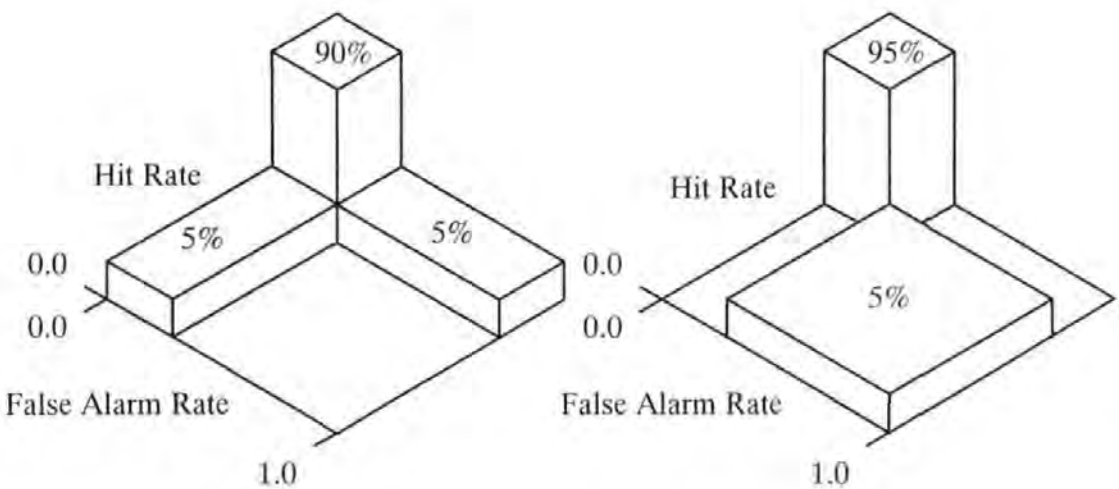


Figure 3.6 Extreme 2D forms of orthogonal 95% posterior intervals

It should also be noted that it is desirable for any point inside the posterior interval to be more likely than any point outside the posterior interval. This is not necessarily true with a rectangular posterior interval constructed this way. A point just inside the corner of the rectangle (e.g. the coordinate $x = 0.18$, $y = 0.26$ marked A in Figure 3.4) could easily have a lower probability than a point just outside the rectangle at a position corresponding

to the peak of either of the two parent probability density functions (e.g the coordinate $x = 0.33$, $y = 0.24$ marked B in Figure 3.4).

There are thus two problems with generating a two dimensional posterior interval from the projection of two one dimensional posterior intervals. Firstly, the probability of the posterior interval can only be given as a range rather than as an exact value, and secondly, the posterior interval may include regions that should be outside, and exclude regions that should be inside. An identical argument applies to Frequentist confidence intervals.

3.2.7 Displaying Confidence and Posterior Intervals

There are other ways of using orthogonal linear confidence and posterior intervals than plotting a rectangle around a ROC point. The most obvious is using them directly in the form of a pair of crossed error bars. This leaves further interpretation to the user. However, the bars can be used as the axes of an ellipse, or even as the coordinates of a parallelogram when the confidence interval has been produced by a logit transformation [44]. If the reasoning discussed in section 3.2.6 is applied to these methods the same problems emerge. This also applies to higher dimensions as in Tilbury et al. [45].

3.2.8 More than Two Categories

Suppose there are now three possible diagnoses – *diseased*, *unknown*, and *healthy* – and that the probability of a healthy patient being diagnosed as *diseased* is x_0, as *unknown* is x_1, and as *healthy* is $x_2 = 1 - x_0 - x_1$; the probability of a diseased patient being diagnosed as *diseased* is y_0, as *unknown* is y_1, and as *healthy* is $y_2 = 1 - y_0 - y_1$. From the discussion in section 3.2 above, it should be clear the probability that a_0 healthy cases are classified as *diseased*, a_1 are classified as *unknown*, a_2 are classified as *healthy*, and that b_0 diseased cases are classified as *diseased*, b_1 diseased cases are classified as *unknown* and b_2 cases are classified as *healthy* is given by:

$$p_{xxyy} \propto x_0^{a_0} \; x_1^{a_1} \; (1 - x_0 - x_1)^{a_2} \; y_0^{b_0} \; y_1^{b_1} \; (1 - y_0 - y_1)^{b_2}$$

In general, if there are n categories the probability, p_n, is given by:

$$p_n \propto \prod_{i=0}^{n-1} x_i^{a_i} \left[1 - \sum_{i=0}^{n-1} x_i\right]^{a_n} y_i^{b_i} \left[1 - \sum_{i=0}^{n-1} y_i\right]^{b_n} \quad \text{where} \quad x_i \geq 0, \quad y_i \geq 0$$

Or alternately, a simpler expression can be used, provided the conditions are stated:

$$p_n \propto \prod_{i=0}^{n} x_i^{a_i} \, y_i^{b_i} \quad \text{where} \quad x_i \geq 0, \quad \sum_{i=0}^{n} x_i = 1, \quad y_i \geq 0, \quad \sum_{i=0}^{n} y_i = 1 \tag{3.4}$$

These equations give Dirichlet distributions and are the foundation of the novel methods presented in Chapter 5.

3.3 Nonparametric AUC for Continuous Data

The Area Under the Curve (AUC) will now be examined. Methods to measure the statistical properties of the nonparametric AUC also involve using permutations as explained in section 3.1.5. The statistical basis was first investigated by Wilcoxon [46] and more thoroughly by Mann and Whitney in 1947 [47]. Bamber [48] pointed out that the Mann–Whitney U statistic, as it is known, is identical to the area under a ROC curve.

The Mann–Whitney model is a Frequentist model that assumes that the measurement taken from the diseased and healthy cases are on a continuous, rather than discrete scale. Given infinite resolution of the continuous scale, ties between two measurements are infinitely unlikely, and are therefore not accommodated in the model. While this has merit, it does produce rather ugly ROC curves for small sample sizes, however Mann & Whitney did not have ROC curves in mind when they did their work; (It was Bamber who noted the connection 28 years later).

Measurement	Diseased	Healthy
1.37	1	0
1.87	1	0
2.45	0	1
3.46	1	0
4.89	0	1
5.43	1	0
6.07	0	1
7.46	0	1

Table 3.1 Continuous ROC data

Figure 3.7 gives the ROC 'curve' of the data in Table 3.1. Each case generates its own category, and therefore every category either has one diseased case and zero healthy cases or visa versa. The data can therefore be represented by a sequence, *y* for one diseased and zero healthy cases and *x* for one healthy and zero diseased cases. For example, the sequence *yyxyxyxx* represents the data in Table 3.1. It can also be interpreted as the instructions to draw the ROC curve – one move in *y* direction; one move in *y* direction; one move in *x* direction; ... . The distribution of the AUC, and hence the Mann–Whitney statistic, is obtained from the distribution of the permutations of this sequence.

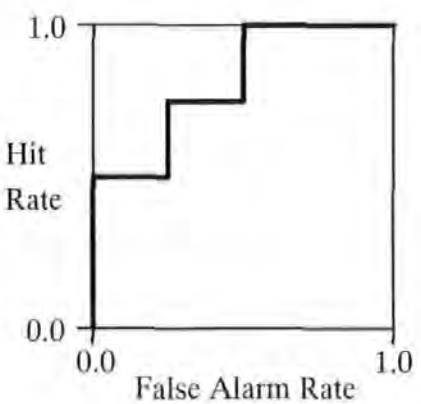


Figure 3.7 ROC graph using continuous data

Graphically, the AUC is obvious. It is just a matter of counting squares. The area can also be calculated directly from the sequence. Start at the end of the sequence and work back. For every *x* encountered, count the number of *y*s preceding it:

yyxyxyx x	*x* preceded by 4 *y*s		area = 4
yyxyxy x	*x* preceded by 4 *y*s	add 4 to area	area = 8

yyxyx	*y*	do nothing		
yyxy	*x*	*x* preceded by 3 *ys*	add 3 to area	area = 11
yyx	*y*	do nothing		
yy	*x*	*x* preceded by 2 *ys*	add 2 to area	area = 13
y	*y*	do nothing		
	y	do nothing		therefore total area = 13

given the grid is 4 × 4, the normalised AUC is 13/16 = 0.8125

The Mann–Whitney U statistic tests if measurements from the sequences of diseased and healthy cases could have come from the same population or not. In ROC terms, it tests if the distributions, e.g. as sketched in Figure 1.1, are the same, or are different. (They are different in Figure 1.1.) The test must therefore make a decision about how unusual the particular sequence of cases is compared with the multitude of sequences that could occur by chance.

If there are n diseased cases and m healthy cases, then, from section 3.1.5, the number of possible sequences is:

$$\frac{(m+n)!}{m!\ n!}$$

However, many of these sequences will give the same AUC. Mann and Whitney devised a recursive function to calculate the number of sequences that produced a given AUC.

If the distributions of the diseased and healthy populations are the same there must be exactly the same probability of generating each different sequence, and therefore the number of sequences that give a particular AUC is proportional to the probability of that AUC. By normalizing the count by dividing by the total number of sequences, the discrete probability density function (pdf) of the AUC can be derived. Mann and Whitney give probability tables for values of n and m up to eight. For values greater than eight the probability density function of the AUC can be approximated by a Gaussian distribution.

Both the power, and limitations of, the Gaussian distribution as an approximation for other distributions is central to the arguments in this thesis, so they will be explained in detail.

3.3.1 The First Central Limit Theorem

Suppose that it is required to calculate the probability of flipping between 500,000 and 501,000 heads out of 1,000,000 flips of a fair coin [49]. From section 3.1.6 the equation is clearly:

$$\sum_{k=500000}^{k=501000} \begin{bmatrix} 1000000 \\ k \end{bmatrix} 0.5^k \ (1 - 0.5)^{1000000-k} \tag{3.5}$$

While the equation is simple, the arithmetic for actually calculating this probability is not. It even presents some difficulty on a computer due to the magnitude of the numbers involved. When this type of problem was first encountered by the pioneers of probability theory while solving problems for their gambling friends, the computational tools were more limited. The problem was solved by de Moivre in 1733 [50] by introducing an approximation now known as the Gaussian, or normal distribution:

$$\begin{bmatrix} a_0 + a_1 \\ a_0 \end{bmatrix} x^{a_0} \ (1 - x)^{a_1} \approx \frac{1}{\sqrt{2\pi(a_0 + a_1)x(1 - x)}} \ e^{-\left(\frac{(a_0 - (a_0 + a_1)x)^2}{2(a_0 + a_1)x(1-x)}\right)}$$

The approximation assumes an infinite number of flips of a fair coin. The fewer the number of flips, and the more biased the coin, the worse the approximation. In practice however, if the chance of a head is given as h, and the chance of a tail is given as t, and n flips are made, the distribution will be approximately Gaussian provided hn and tn are both greater than five.

If $\mu = (a_0 + a_1)\ x$ and $\sigma^2 = (a_0 + a_1)\ x\ (1 - x)$ the equation takes its more familiar form:

$$\frac{1}{\sqrt{2\pi\sigma^2}} \ e^{-\frac{(a_0 - \mu)^2}{2\sigma^2}}$$

The Gaussian distribution is fully characterised by these two parameters, the mean, μ, and the standard deviation, σ.

If the mean, μ, is 0, and the standard deviation, σ, is 1, the standard normal distribution, is obtained:

$$\frac{1}{\sqrt{2\pi}} e^{-\frac{a_0^2}{2}} \tag{3.6}$$

(For the solution to equation 3.5, see Stirzaker [49].)

3.3.2 Standard Deviation of the Mann–Whitney U Test

Mann and Whitney proved that the distribution of the U statistic is asymptotic at the limit when n and m are large (the same criteria for proving the binomial distribution converges to a Gaussian). In order to apply the Gaussian distribution to the Mann–Whitney U statistic the distributions need to be aligned. This can be achieved by finding the mean and standard deviation of both distributions, translating so the means are identical, and scaling so the standard deviations are identical.

Mann and Whitney used the recursive function for the distribution of the AUC to derive a formula for the standard deviation, σ:

$$\sigma = \sqrt{\frac{n + m + 1}{12nm}}$$

Suppose an intelligent medical system was tested on 50 diseased cases and 60 healthy cases and gave a diagnosis in the form of a real number between 0 and 1. The Mann–Whitney U statistic can be used to determine if the system has discriminated between the two groups. The null hypothesis is that there is no difference, i.e. as far as the system is concerned, the cases come from the same population, and the AUC is 0.5. The alternate hypothesis is that the populations are different, and therefore that the system can discriminate between them.

$$\sigma = \sqrt{\frac{50 + 60 + 1}{12 \times 50 \times 60}} = \sqrt{\frac{111}{36000}} = \sqrt{0.0030833} = 0.0555$$

The 95% confidence interval of the Gaussian distribution is (by definition) $\pm$ 1.96 standard deviations from the mean. Therefore, if the observed AUC is outside the range $0.5 \pm 1.96 \times 0.0555 = 0.5 \pm 0.1088$ then the null hypothesis, that the populations are the same (as far as the system is concerned), should be rejected.

3.3.3 The Triangular Distribution

However, this calculation hangs critically on the assumption that the AUC has a Gaussian distribution. But, just to illustrate a point, suppose this is false, and the distribution is in fact the symmetric triangular distribution as shown in Figure 3.8.

The mean, μ, and standard deviation, σ, can be calculated for any distribution, and are defined (for a discrete distribution) by the formulae:

$$\mu = \sum_{x \in \Omega} x\ p(x) \tag{3.7}$$

$$\sigma = \sqrt{\sum_{x \in \Omega} (x - \mu)^2\ p(x)} \tag{3.8}$$

Where Ω is the set of possible values of x, and $p(x)$ is the probability of a given value of x.

For a continuous distribution the sum becomes an integral. The standard deviation, σ, of the symmetric triangular distribution is therefore:

$$\sigma = \frac{1}{\sqrt{6}}$$

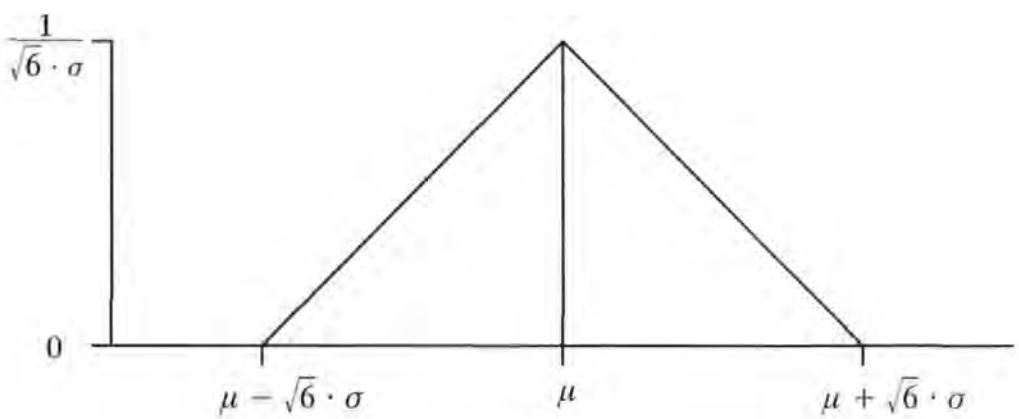


Figure 3.8 Triangular distribution

The 95% confidence interval, assuming a triangular distribution, is now $0.5 \pm \left(1 - \sqrt{0.05}\right) \times \sqrt{6} \times \sigma \ = 0.5 \pm 1.90 \times 0.0555 \ = 0.5 \pm 0.1056$ (which is actually rather close to the real value).

The point is that knowing the mean and the standard deviation of the distribution is not enough to determine the confidence interval, the form of the distribution is required as well. Given how easy it is to calculate μ and σ for any distribution (equations 3.7 and 3.8), this is frustrating. One solution is to use a distribution that is wider than the real distribution.

3.3.4 Interpretation of a Confidence Interval

If the only concern is the proportion of a long run of samples being inside the confidence interval and there is no concern about the proportion of a long run of samples being outside, then a 'safe bet' can be taken and either the confidence interval can be made larger, or the proportion smaller. Hence, under this interpretation, one statistical method can generate a 95% confidence interval twice as big as another, and they can both be correct.

If the proportion of a population point falling outside the interval is also required, then a far more pedantic position has to be taken. The proportion of a population point falling in the confidence interval has to be precisely and exactly the figure claimed, and hence the proportion of the population falling outside has to be one minus the proportion claimed.

For the purposes of this thesis the former will be called a conservative confidence interval, the latter an exact confidence interval.

3.3.4.1 Definition of an Exact Confidence Interval

The $z\%$ confidence interval of the population variable P, is the minimum range of values of P that contains exactly $z\%$ of a theoretically long run of samples drawn using the estimated value of the population variable p.

3.3.4.2 Definition of a Conservative Confidence Interval

The $z\%$ confidence interval of the population variable P, is the minimum range of values of P that has $c\%$, where $100\% \leq c \leq z\%$, of a theoretically long run of samples drawn using the estimated value of the population variable p.

3.4 Nonparametric AUC for Discrete Data

If the data is discrete, rather than continuous, the possibility of ties is introduced. If the data in Table 3.1 is rounded to the nearest integer, the resulting discrete data has two ties. Table 3.2 shows this discrete version, which gives the ROC curve in Figure 3.9.

Measurement	Diseased	Healthy
1	1	0
2	1	1
3	1	0
5	1	1
6	0	1
7	0	1

Table 3.2 Discrete ROC data

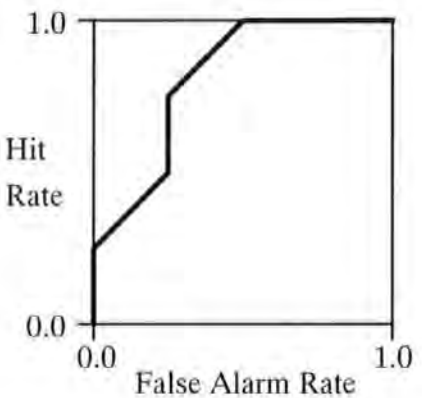


Figure 3.9 ROC graph using discrete data

In order to calculate the area from the raw data, a function, $\psi(x,y)$, is defined:

$$\psi(x,y) = \begin{cases} 1.0 & x < y \\ 0.5 & x = y \\ 0.0 & x > y \end{cases}$$

The Area Under the Curve is then given by:

$$AUC = \frac{1}{mn} \sum_{j=1}^{j=n} \sum_{i=1}^{i=m} \psi(X_i, Y_j) \tag{3.9}$$

The data in Table 3.2 is then represented by the vectors:

$X = [\ 2\ 5\ 6\ 7\ ]$ $\qquad$ $Y = [\ 1\ 2\ 3\ 5\ ]$

The standard deviation is then given by:

$$\sigma = \sqrt{\frac{\sum_{i=0}^{i=m}\left(\left[\frac{\sum_{j=0}^{j=n}\psi(X_i,Y_j)}{n}\right] - AUC\right]^2}{m(m-1)} + \frac{\sum_{j=0}^{j=n}\left(\left[\frac{\sum_{i=0}^{i=m}\psi(X_i,Y_j)}{m}\right] - AUC\right]^2}{n(n-1)}} \tag{3.10}$$

There are up to four different methods of calculating the standard deviation in the literature:

1. A nonparametric AUC was first suggested by Bamber [48], and subsequently simplified by Hanley and McNeil [51], though they did not show how it was derived.

2. A second simplified method was given by Hanley and McNeil [51].

3. DeLong et al. [52] produced a third method in their search for the covariance between two AUCs (according to Hanley and Hajian–Tilaki[53]), though DeLong et al. claim it can be shown to be equivalent to Bamber's method.

4. Methods for calculating the variance using jackknifing have been devised by McNeil and Hanley [54], and extended by Dorfman et al.[55] Jackknifing will not be examined further.

Equation 3.10 is Obuchowski and Lieber's [19] version of DeLong et al.'s method [52], as used in their Monte Carlo experiments explained in section 3.6.1.

For a standard deviation to be any use in generating confidence intervals, the probability density function needs to be known as well. In a paper on general U statistics, Hoeffding [56], proved that the distribution is Gaussian, whatever the AUC, at infinite sample size. Both Bamber [48] and DeLong et al. [52] use this result, though Bamber ponders the problem of small sample sizes, and concludes he can offer no guidance as to the sample size at which the Gaussian becomes a good approximation for the distribution.

3.4.1 The Trapezoid Rule

If there are many cases in each category, as in Table 3.3, a ROC 'curve' of straight line segments can be plotted as in Figure 3.10.

Measurement	Diseased	Healthy
1	4	2
2	2	2
3	2	4

Table 3.3 Categorical ROC data

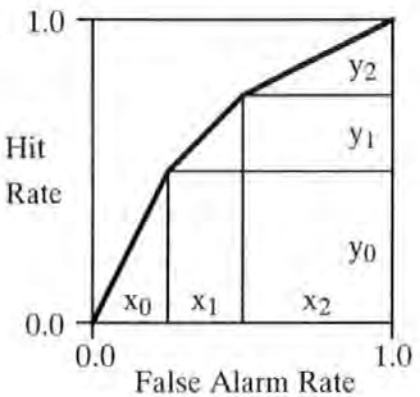


Figure 3.10 ROC curve using categorical data

Geometrically, the AUC is therefore given by the trapezoid rule:

$$AUC = \frac{x_0 y_0}{2} + \frac{x_1(y_0 + y_0 + y_1)}{2} + \frac{x_2(y_0 + y_0 + y_1 + y_1 + y_2)}{2}$$

This gives exactly the same result as the calculation of the AUC by equation 3.9, where:

$\boldsymbol{X}$ = [1 1 2 2 3 3 3 3] $\quad\quad$ $\boldsymbol{Y}$ = [1 1 1 1 2 2 3 3]

The larger the number of categories, and the larger the number of cases in each category, the more the plot will approach a smooth curve. However, even for a small sample size, it might be suspected that the population ROC curve is actually smooth, and it is desirable to be able to estimate the curve.

3.5 The Parametric ROC Model

In order to plot a smooth ROC curve for a small sample, a parametric model of the data needs to be used. The most common model is to assume that the distributions of both the diseased and healthy populations are Gaussian.

To be strictly correct, the binormal ROC model only requires the diseased and healthy populations to be latently Gaussian. A latently Gaussian distribution is any distribution which gives a Gaussian distribution after a monotonic transformation. A monotonic trans-

formation preserves the relational operators > and < between any pair of values. For example, if blood pressure was being measured in diseased and healthy patients and used to plot a ROC curve, the square root of blood pressure would be found to produce an identical ROC curve.

Two parameters specify the shape of the ROC curve – the difference between the means, and the ratio of their standard deviations. These two parameters are fully determined by two ROC points. Figure 3.11 gives a graphical explanation of how the parameters can be derived from these two points.

Graph (1), at the top left of Figure 3.11, shows two ROC points joined by a line. The projection of the Hit Rate, or y, coordinates of the points (right), and the False Alarm Rate, or x, coordinates (down), onto the cumulative standard normal distribution (graphs 2a and 2b respectively) gives the threshold values of each point in the healthy and disease populations. The cumulative standard normal distributions are scaled to give the same difference of threshold, (graph 3a and 3b) and translated so the thresholds coincide (graph 4), where they are plotted as normal distributions (rather than cumulative normal distributions), to give the underlying parametric model for the ROC curve. (Graphs 3a and 3b also show the normal distributions, drawn in dotted lines, for easy comparison).

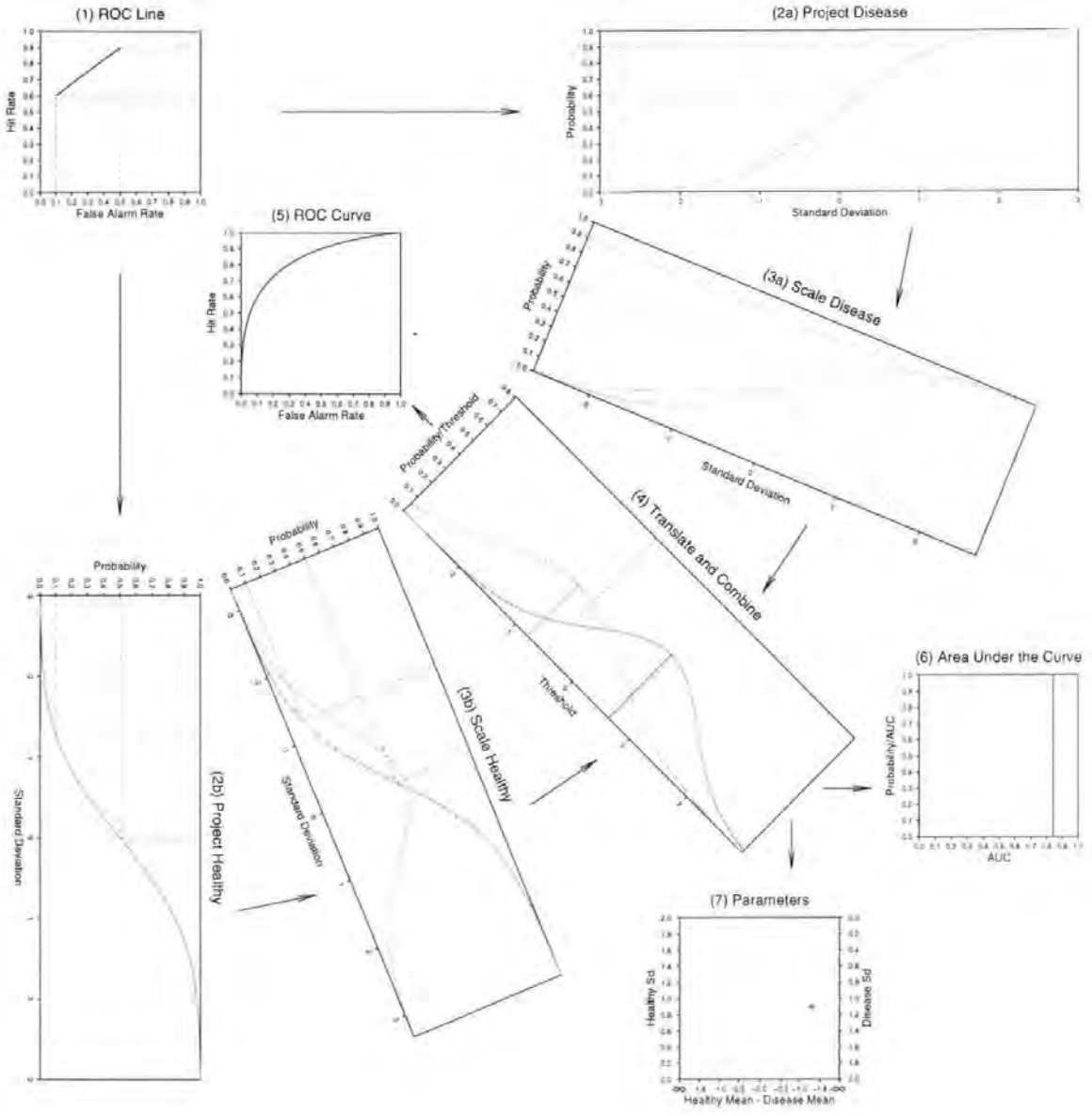


Figure 3.11 Explanation of binormal ROC curves

Once the underlying model is derived, the parametric ROC curve can be plotted (graph 5), and the AUC calculated (graph 6). It is also possible to plot the parameters as a point on a graph of the difference of mean against the standard deviation (graph 7), provided appropriate scales are used.

Algebraically, the difference in standard deviation of the healthy and diseased populations ($\Delta\sigma_x$ and $\Delta\sigma_y$ respectively) is given by:

$$\Delta\sigma_x = \Phi^{-1}(x_0 + x_1) - \Phi^{-1}(x_0)$$

$$\Delta\sigma_y = \Phi^{-1}(y_0 + y_1) - \Phi^{-1}(y_0)$$

Where Φ^{-1} has the definition:

if $p = \Phi(z)$ then $z = \Phi^{-1}(p)$

where $\Phi(z)$ is the cumulative standard normal distribution (integral of equation 3.6):

$$\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-\frac{x^2}{2}}\, dx$$

and where

x_0 is the start False Alarm Rate coordinate of the line

y_0 is the start Hit Rate coordinate of the line

$x_0 + x_1$ is the end False Alarm Rate coordinate of the line

$y_0 + y_1$ is the end Hit Rate coordinate of the line

In order to scale the differences of standard deviations, the following scaled standard deviations will be defined for the healthy, σ_h, and diseased, σ_d, populations:

$$\sigma_h = \frac{2 \cdot \Delta\sigma_x}{\Delta\sigma_x + \Delta\sigma_y}$$

$$\sigma_d = \frac{2 \cdot \Delta\sigma_y}{\Delta\sigma_x + \Delta\sigma_y}$$

These definitions allows either σ_h or σ_d to be zero without causing a division by zero, (if both are zero, there aren't two points to specify the curve) and for the standard deviations to be 1.0 when $\sigma_h = \sigma_d$.

The difference in mean, $\Delta\mu$, can now be defined as:

$$\Delta\mu = \frac{\Phi^{-1}(x_0)\sigma_d - \Phi^{-1}(y_0)\sigma_h}{2}$$

or, equivalently:

$$\Delta\mu = \frac{\Phi^{-1}(x_0 + x_1)\sigma_d - \Phi^{-1}(y_0 + y_1)\sigma_h}{2}$$

Since $\Delta\mu$ is in the range $\pm\ \infty$, a sigmoid function is used for plotting in the finite range $\pm\ 1$:

$$\Delta\mu' = \frac{1}{(1 + e^{\Delta\mu f})}$$

Where f is a scaling factor, introduced to give a more easily visualised scale to the graph, such that:

when $\Delta\mu = -1$ then $\Delta\mu' = -0.5$ and

when $\Delta\mu = +1$ then $\Delta\mu' = +0.5$

(The value of f is 1.0986 to 4 d.p.)

Graph 7 (of Figure 3.11) is thus a plot of $\Delta\mu'$ against σ_h (or $2 - \sigma_d$), with the horizontal scale marked in units of $\Delta\mu$. A binormal ROC curve is represented by a single point on this plot. The ROC curve in Graph 5 (of Figure 3.11) is thus completely specified by the point marked with a cross in Graph 7. This parsimonious representation gives an alternative way of visualising binormal ROC curves.

3.5.1 Least Squares Fit

While the model is quite straight forward, the actual problem is to calculate the best ROC curve given a set of more than two data points. The easiest solution is to plot the data points on Gaussian–Gaussian graph paper [57], and fit a line to the points either by eye [58], or by using a least squares fit. The same result can be calculated directly without resorting to graph paper. It should be noted that for this application a least squares fit has no more theoretical justification than fitting by eye.

3.5.2 Maximum Likelihood

A more sophisticated, and theoretically justifiable, solution is to use a least squares fit as a start point, and then use a hill climbing algorithm to iterate to the maximum likelihood solution. The maximum likelihood is the value of the parameters giving the greatest probability to the observed event. Kendell et al. give a formal definition [59]. This method was first applied by Dorfman and Alf [58][60], and subsequently developed by Metz et al. [61].

From equation 3.4, the probability of a point is given by:

$$p_n \propto \prod_{i=0}^{n} x_i^{a_i} \, y_i^{b_i} \qquad \text{where} \quad x_i \geq 0, \quad \sum_{i=0}^{n} x_i = 1, \quad y_i \geq 0, \quad \sum_{i=0}^{n} y_i = 1$$

Defining x_i and y_i in terms of the cumulative standard normal distribution:

$$x_i = \Phi(z_i) - \Phi(z_{i-1}), \qquad y_i = \Phi(\beta z_i - \alpha) - \Phi(\beta z_{i-1} - \alpha)$$

Where:

$$\alpha = \frac{(\mu_d - \mu_h)}{\sigma_d}, \qquad \beta = \frac{\sigma_h}{\sigma_d}, \qquad z_i = \frac{(t_i - \mu_h)}{\sigma_h}$$

and where:

μ_d is the mean of the disease sample

μ_h is the mean of the healthy sample

σ_d is the standard deviation of the disease sample

σ_h is the standard deviation of the healthy sample

t_i is the i^{th} threshold

Then:

$$p_n \propto \prod_{i=0}^{n} (\Phi(z_i) - \Phi(z_{i-1}))^{a_i} \, (\Phi(\beta z_i - \alpha) - \Phi(\beta z_{i-1} - \alpha))^{b_i} \qquad (3.11)$$

Where:

$$\Phi(z_{-1}) = 0, \qquad \Phi(z_n) = 1, \qquad \Phi(\beta z_{-1} - \alpha) = 0, \qquad \Phi(\beta z_n - \alpha) = 1$$

because threshold $t_{-1} = -\infty$ and $t_n = +\infty$.

Taking natural logs:

$$\ln p_n \propto \sum_{i=0}^{n} a_i \ \ln(\Phi(z_i) - \Phi(z_{i-1})) + b_i \ \ln(\Phi(\beta z_i - \alpha) - \Phi(\beta z_{i-1} - \alpha))$$

The Newton–Raphson method can be used to find the maximum of this function. An initial vector V of the parameters, α, β, z_0, ... z_{n-1}, of the curve is set up. These initial values can be calculated by a least squares fit as discussed in section 3.5.1. The initial estimate, V, is updated iteratively by subtracting the product of the inverse of the Hessian matrix

$\boldsymbol{H}$ (the matrix of second derivatives) of the curve, and the slope of the estimates about the curve $\nabla \boldsymbol{V}$ (the first derivatives):

$$\boldsymbol{V}_{k+1} = \boldsymbol{V}_k - \boldsymbol{H}_k^{-1} \cdot \nabla \boldsymbol{V}_k$$

Where, k is the iteration, and, where:

$$\boldsymbol{V} = \begin{bmatrix} \alpha \\ \beta \\ z_0 \\ \cdots \\ z_{n-1} \end{bmatrix} \quad \boldsymbol{H} = \begin{bmatrix} \frac{\delta lnP'}{\delta \alpha^2} & \frac{\delta lnP'}{\delta \beta \alpha} & \frac{\delta lnP'}{\delta z_0 \alpha} & \cdots & \frac{\delta lnP'}{\delta z_{n-1} \alpha} \\ \frac{\delta lnP'}{\delta \alpha \beta} & \frac{\delta lnP'}{\delta \beta^2} & \frac{\delta lnP'}{\delta z_0 \beta} & \cdots & \frac{\delta lnP'}{\delta z_{n-1} \beta} \\ \frac{\delta lnP'}{\delta \alpha z_0} & \frac{\delta lnP'}{\delta \beta z_0} & \frac{\delta lnP'}{\delta z_0^2} & \cdots & \frac{\delta lnP'}{\delta z_{n-1} z_0} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\delta lnP'}{\delta \alpha z_{n-1}} & \frac{\delta lnP'}{\delta \beta z_{n-1}} & \frac{\delta lnP'}{\delta z_0 z_{n-1}} & \cdots & \frac{\delta lnP'}{\delta z_{n-1}^2} \end{bmatrix} \quad \nabla \boldsymbol{V} = \begin{bmatrix} \frac{\delta lnP'}{\delta \alpha} \\ \frac{\delta lnP'}{\delta \beta} \\ \frac{\delta lnP'}{\delta z_0} \\ \cdots \\ \frac{\delta lnP'}{\delta z_{n-1}} \end{bmatrix}$$

When $\nabla \boldsymbol{V}$ is 0, the curve is at a maximum, (it could also be at a minimum or saddle point), and the inverse Hessian matrix, $\boldsymbol{H}^{-1}$, gives the variances and covariances of the curve parameters [62]. The distributions are assumed to be Gaussian for large sample sizes.

The AUC is given by [63]:

$$AUC = \Phi\left(\frac{\alpha}{\sqrt{(1+\beta^2)}}\right)$$

Note, that if the standard deviation of the healthy sample is 0, β is 0, and if the standard deviation of the diseased sample is 0, then β is ∞. It was in order to prevent this problem that different parameters were used in section 3.5. This is a subtle but significant difference between the parameters used in section 3.5 and the parameters found in the literature.

3.5.3 Degenerate Parametric Samples

If the sample size is small, or if the test is very accurate, there is a real possibility that the sample will have zero false positives or zero false negatives. In fact, with an intelligent system, this is what is hoped for, but what is good news for the system designers is bad

news for maximum likelihood ROC analysis. Zero cases implies the threshold is at minus infinity, which prevents the maximum likelihood algorithm from converging. Such data sets are termed 'degenerate', and have to be discarded from analysis.

3.6 Monte Carlo Experiments

3.6.1 Small Sample ROC Failure

Both parametric and nonparametric AUC confidence intervals are calculated using the Gaussian distribution. This is only valid when the sample size is large. Obuchowski and Lieber [19] ran a series of Monte Carlo experiments to determine how large the sample has to be for the methods to be valid. They used up to seven different methods, for both continuous and discrete data, and for both the AUC of one ROC curve, and the difference between the area of two curves.

Tables 3.4 to 3.7 show their results for DeLong's [52] nonparametric test (section 3.4), and for Metz's parametric maximum likelihood program ROCFIT [64] (section 3.5.2), for discrete data with an AUC of either 0.8 or 0.95.

Each Monte Carlo simulation generated 1,000 random test data sets drawn from binormal distributions with equal standard deviations ($\beta = 1$) where the AUC was either 0.8 or 0.95, with sample sizes of between 10 and 70 healthy cases (columns, marked 'H') and between 10 and 70 diseased cases (rows, marked 'D'). The number of discrete categories was not reported.

Tables 3.4 to 3.7 shows the percentage of test data sets whose 95% AUC confidence interval included the AUC used to generate the test. With 1,000 test sets in each run there was a 86% chance that at least 93% of the tests would be in the 95% confidence interval, and a 99% chance that 92% of the tests would be in the 95% confidence interval. Tables 3.4 to 3.7 highlights those results below 92% in grey, and those between 92% and less than 93% in light grey. Obuchowski and Lieber assumed conservative confidence intervals and do not comment on tests results above 97% or 98%. Tables 3.4 to 3.7 also highlights them, so the relevant confidence intervals should be 72% and 98% respectively.

Given 100 table entries in total, about 2 grey (98%), 26 light grey and 72 white entries could be expected. There are 48 grey, 13 light grey and 39 white entries, distributed with systematic bias indicating the performance is worse the smaller the sample size.

Unfortunately these results are not quite what they seem. The next section explains.

DeLong's nonparametric test on discrete data generated from an AUC of 0.8					
D \ H	10	20	30	50	70
10	90.9	90.7	90.8	89.5	87.8
20	90.1	91.6	93.8	93.4	92.1
30	89.0	93.3	94.7	94.1	93.6
50	89.0	91.8	94.3	93.5	95.0
70	88.5	92.7	94.4	95.5	94.5

Table 3.4 Results for AUC of 0.8 on nonparametric ROC

DeLong's nonparametric test on discrete data generated from an AUC of 0.95					
D \ H	10	20	30	50	70
10	85.3	86.0	81.6	78.6	77.4
20	85.7	89.0	87.4	87.2	85.3
30	83.1	86.8	89.8	88.5	90.3
50	80.0	87.0	91.5	90.8	91.7
70	75.2	85.7	88.9	93.9	93.2

Table 3.5 Results for AUC of 0.95 on nonparametric ROC

Metz's parametric test on discrete data generated from an AUC of 0.8					
D \ H	10	20	30	50	70
10	92.0	92.6	93.3	91.3	90.8
20	92.7	92.8	93.6	93.7	92.0
30	92.2	94.0	93.7	93.6	93.7
50	91.8	92.3	94.1	93.8	94.9
70	91.1	92.8	93.9	95.7	94.4

Table 3.6 Results for AUC of 0.8 on parametric ROC

Metz's parametric test on discrete data generated from an AUC of 0.95					
D \ H	10	20	30	50	70
10	99.2	98.5	98.7	98.6	98.0
20	91.6	96.9	96.2	96.1	96.7
30	89.9	93.1	94.7	94.9	94.7
50	90.5	91.3	94.2	94.0	95.1
70	86.9	92.9	91.4	94.8	95.0

Table 3.7 Results for AUC of 0.95 on parametric ROC

3.6.2 Design of Monte Carlo Experiments

Monte Carlo experiments have to be carefully designed in order to actually test a given hypothesis.

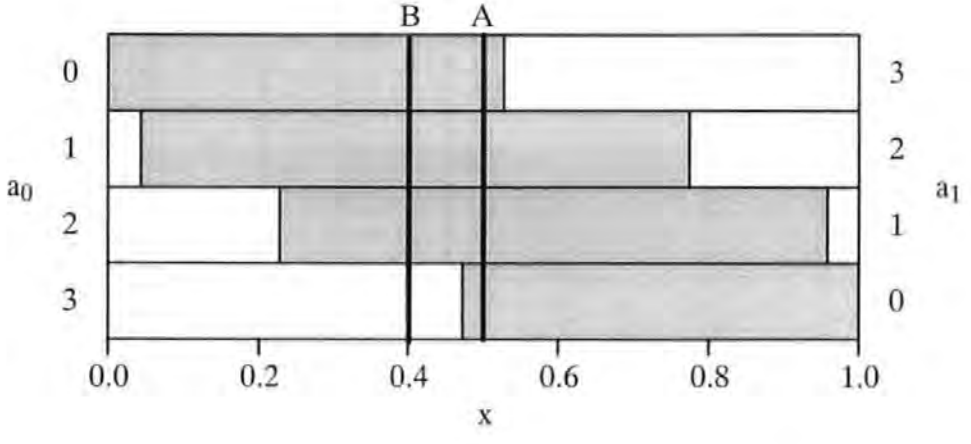


Figure 3.12 Confidence intervals of a small sample

Suppose a Monte Carlo simulation is required to test the 95% posterior interval, as given by the beta distribution (section 3.2), of three flips of a coin. This experiment of three flips will be repeated E times, where, for the purposes of this explanation, E approaches infinity. (For a real simulation the value of E would be as large as possible within the restriction of the time and effort required to simulate each experiment). Each experiment will result in a_0 heads and a_1 tails. The only possible results of each experiment are therefore the a_0:a_1 pairs 0:3, 1:2, 2:1, 3:0. Thus there are only four possible beta functions, and four possible 95% confidence intervals that can possibly be generated. These four 95% confidence intervals are illustrated in Figure 3.12.

Suppose that for each experiment the probability of flipping a head is set to 0.5, and then the three coin flips are simulated. All four possible 95% posterior intervals include 0.5 in their range, (shown as the bold vertical line marked A on Figure 3.12), so after E flips, E experiments are within the posterior interval. The 95% posterior interval looks like a 100% posterior interval.

Now suppose the probability of a head is 0.4. There is a probability of $0.4^3 = 0.064$ of flipping three heads. However, the probability of 0.4 (line B on Figure 3.12) is now outside the 95% posterior interval obtained from flipping three heads. Therefore, after E flips it would be expected that only 93.6% of the experiments are within the 95% posterior interval. It would now appear that the posterior interval is too small.

The effect has nothing to do with the particular distribution the posterior interval (or confidence interval) is calculated from, but is caused by the discrete nature of the experiment. If a coin is flipped n times, there are only $n + 1$ different results and therefore only $n + 1$ different posterior or confidence intervals. Setting the probability of a head to any given value has the inevitable effect of including some posterior/confidence intervals and excluding others.

The smaller n is the more noticeable this effect becomes. The effect is investigated in a Monte Carlo experiment discussed in Section 6.3. For now it should be noted that Obuchowski and Lieber's experiment [19] is limited as a consequence of the Frequentist paradigm, along with many other studies on ROC curve statistics in the literature [44] [53] [63] [65] [66] [67] [68] [69] [70] [71] [72] [73] [74] [75].

3.7 Summary

This chapter has examined the main statistical methods of ROC curve analysis currently in the literature. They are all based on the asymptotic approach of the distribution of the statistics to the Gaussian distribution with increasing sample size.

While the validity of these statistics is beyond question as the sample size tends to infinity, the same is not true when the sample size tends to zero. Given the nature of intelligent medical system evaluation it is unfortunate that it is at this limit that statistics are required.

It is also unfortunate that the current Monte Carlo simulations used to test the limit of validity of existing statistics are themselves theoretically limited at small sample sizes. It leaves the precise limit to how low the sample size can fall before the approximation becomes unacceptable totally unknown.

Mann and Whitney generated the pdf for their statistic for up to eight healthy and eight diseased cases and found it was Gaussian to an acceptable approximation for higher sample sizes. However, this distribution only applies when the AUC is exactly 0.5. (Further implications of this are left to be discussed in section 8.1.1, after novel methods for generating a better solution have been explained in section 5.4) What happens when the AUC is not 0.5 does not appear to be known, but from the limitation of the binomial approximation of the Gaussian, it can safely be assumed that the sample size for an acceptable Gaussian approximation is higher. It might be argued that no one would want to test an intelligent medical system with only eight disease cases. However, as discussed in section 7.1, Adlassnig and Scheithauer [10] did precisely that.

The concept of generality should not be forgotten either. Obuchowski and Lieber's could not recommend a single best statistic to use out of all the seven methods they tried. They gave a table of different tests that performed best under a variety of different circumstances. This is awkward and frustrating for anyone who needs to use ROC curves as a tool to get a job done. Given the discussion on designing Monte Carlo experiments it is doubtful if this table is even valid.

This leaves a problem. If a new intelligent medical system is ready for a small pilot clinical trial and ROC curves would provide an ideal evaluation format, there is nowhere to turn for the statistical computations. It was in order to solve this problem that novel methods were investigated. They are presented in Chapter 5, after existing statistics for confusion matrixes are presented in Chapter 4.

4 Confusion Matrix Statistics

This chapter examines existing methods for statistical analysis of confusion matrices. Only one method, kappa, would appear to have any merit. Methods used prior to kappa being introduced are discussed in order to explain the advantage of the kappa statistic in accounting for chance correction. The kappa statistic has made the chi-squared statistic redundant for confusion matrices, but chi-squared is ideal in other fields. It is the test used in all the Monte Carlo experiments that validate the novel methods presented in this thesis.

4.1 Simple Agreement

An example of data presented as a confusion matrix can be seen in Figure. 4.1 below (after Cohen [13]). There are two numbers in each cell, the first is the observed number of cases, the second, in parentheses, is the number expected by chance, calculated by multiplying the total for the row, by the total for the column, and dividing by the total of the table. Given a confusion matrix like this, how can agreement be measured?

The most primitive approach to measuring agreement is simply to count the proportion of cases that fall along the diagonal, in this instance, 29%, and use that as an agreement measure. This does not take account of chance. Here, the expected score by chance alone is 35%.

		'Gold Standard'			
		A	B	C	Row Sum
Test	A	50 (40)	26 (30)	24 (30)	100
	B	24 (24)	4 (18)	16 (18)	60
	C	6 (16)	30 (12)	4 (12)	40
Column Sum		80	60	60	200

Figure 4.1 Example confusion matrix

4.2 Chi–Squared

To correct for chance, many investigators have used the chi–squared statistic (χ^2), over all cells, to test the hypothesis that the observed frequencies are produced by chance alone:

$$\chi^2 = \sum_{n=1}^{n=N^2} \frac{(Observed_n - Expected_n)^2}{Expected_n}$$

Where N is the number of rows (or columns) of the confusion matrix. In the case of Figure 4.1 above the chi–squared is:

$$\frac{(50-40)^2}{40} + \frac{(26-30)^2}{30} + \frac{(24-30)^2}{30} + \frac{(24-24)^2}{24} + \frac{(4-18)^2}{18} + \frac{(16-18)^2}{18} + \frac{(6-16)^2}{16} + \frac{(30-12)^2}{12} + \frac{(4-12)^2}{12} = 64.59$$

Which is a highly significant result for 4 degrees of freedom. As a test to determine if the data are produced by chance alone, chi–squared is perfectly valid. However, as Cohen [13] pointed out, as a test of agreement, the test is invalid. This can be seen from the large contribution to the chi–squared result provided by cell *Gold B, Test C*, which is in fact a disagreement:

$$\frac{(30-12)^2}{12} = 27.0$$

Therefore chi–squared is a measurement of association, not agreement.

4.3 Kappa

Cohen [13] proposed the kappa statistic for measuring agreement in confusion matrices. The statistic only examines the agreement diagonal of the matrix, and compares the observed agreement against that expected by chance. The kappa statistic is defined as:

$$K = \frac{p_o - p_c}{T - p_c}$$

$$Where \qquad p_o = \sum_{n=1}^{n=N} Observed_{nn} \quad and \quad p_c = \sum_{n=1}^{n=N} Expected_{nn}$$

N is the number of rows (or columns) of the confusion matrix, and T the total number of cases in the table.

The kappa for Figure 4.1, is therefore:

$$\frac{(50 + 4 + 4) - (40 + 18 + 12)}{200 - (40 + 18 + 12)} = \frac{58 - 70}{200 - 70} = \frac{-12}{130} = -0.0923$$

As it has already been noted, the proportion of cases agreed on (29%) is less than expected by chance (35%), so it is not surprising that a negative agreement statistic is produced. A highly significant chi-squared only means there is very significant association between the test and the 'Gold Standard'. In this instance, the association is a disagreement.

Note, that if there is perfect agreement, p_o will be T, and therefore kappa will be 1. If $p_o = p_c$, the agreement is precisely that expected by chance, and kappa will be 0. In the extreme case kappa can be −1. Landis and Koch [76] have suggested an arbitrary linguistic classification for kappa values:

Kappa Statistic	Strength of Agreement
≤0.00	Poor
>0.0 to ≥ 0.2	Slight
>0.2 to ≥ 0.4	Fair
>0.4 to ≥ 0.6	Moderate
>0.6 to ≥ 0.8	Substantial
>0.8 to ≥ 1.0	Almost Perfect

Table 4.1 Linguistic variables for kappa

Cohen [13] also gave an expression for the standard error:

$$\sigma = \sqrt{\frac{p_o(1 - p_o)}{N(1 - p_c)^2}}$$

and noted that the distribution of kappa is approximately normal for a large sample size.

4.4 Weighted Kappa

The kappa statistic, as given above, assumes that only the perfect matches are valid, all other combinations are equally invalid. However, assigning clinical effectiveness measures to decisions may show a more complex picture. Some mismatches might be classed as acceptable near misses, having nearly the same clinical effectiveness as perfect matches, others might have dire consequences, and appropriately punitive clinical effec-

tiveness measures. Cohen [14] developed a version of kappa, weighted kappa, where each cell in the matrix is given a weight indicative of its relative value:

$$K_w = \frac{p_o - p_c}{T - p_c}$$

Where:

$$p_o = \sum_{n=1}^{n=N} \sum_{m=1}^{m=N} Observed_{nm} \cdot Weight_{nm}$$

$$p_c = \sum_{n=1}^{n=N} \sum_{m=1}^{m=N} Expected_{nm} \cdot Weight_{nm}$$

N is the number of rows (or columns) of the confusion matrix, and T the total number of cases in the table.

It is convenient to assign the weights so that the maximum weight is 1, assigned to the perfect agreement diagonal, and weights between 0 and 1 are assigned to the disagreements, in a ratio proportional to the degree of agreement. A weight of 0.5 is twice the agreement of a weight of 0.25, and therefore ratio measures of clinical effectiveness can be used. The weight matrix is also important in calculating the significance of the test, as discussed by Cohen:

$$\sigma \approx \sqrt{\frac{\sum_{n=1}^{n=N} \sum_{m=1}^{m=N} Weight_{nm}^2 Observed_{nm} - {p_o}^2}{N(1 - p_c)^2}}$$

Cohen stated that the standard error has a Gaussian distribution at large sample sizes.

4.5 Testing Kappa Confidence Intervals

Exactly the same limitations have been made in Monte Carlo investigations of the confidence interval of the kappa statistic as have been made with ROC curves. These limitations apply equally to investigations of the general case of confusion matrices and to particular forms.

4.5.1 The Two by Two Case

The simplest confusion matrix is a two by two grid, known as the '2 × 2 case'. It is identical in format to a ROC single threshold contingency table (Table 1.1). Kappa is one of many non–equivalent methods for measuring agreement of the 2 × 2 case [77].

Blackman and Koval [77] investigated four different methods for generating the 95% confidence interval of the kappa statistic for the 2 × 2 case. All four methods (Bloch and Kraemer [78], Fleiss et al. [79], Garner [80], and a jackknifing method developed successively by Quenouille [81], Tukey [82] and Miller [83]), calculate the variance of kappa, and derive the confidence interval by assuming a Gaussian distribution.

Blackman and Koval used exact computation, rather than simulation, to test the accuracy of the 95% confidence interval. They tested all 250 combinations of the sample sizes 20, 30, 40, 50, 60, 80, 100, 120, 150, 200, the population kappa values 0.0, 0.2, 0.4, 0.6, 0.8, and the population frequencies of disease 0.1, 0.2, 0.3, 0.4, 0.5. The frequency of disease had exactly the same meaning as it would for the ROC data in Table 1.1.

For each of the 250 combinations, every possible 2 × 2 matrix was generated for the respective sample size. The probability of each matrix was calculate from the respective population kappa value and population frequency of disease. The sample kappa value, sanple variance, and hence 95% confidence interval, was calculated for each matrix. The number of times the sample kappa value was inside the estimated population 95% confidence interval was recorded. It was expected that 95% of the sample kappa values would be within the 95% confidence interval of the estimated population.

This was not what was found. The lower the sample size, the further the results were from the expected value of 95%. The explanation for this observation has already been given in section 3.6.2. Blackman and Koval's experiment appears to be a scaled up version of the coin flipping experiment. If it is, no conclusions can be drawn about the validity of the kappa statistics.

4.5.2 The General Case

In a recent paper, Altaye et al. [20] proposed a new confidence interval for kappa, for the general $n \times n$ case. They imply their method is accurate for smaller sample sizes than exist-

ing kappa statistics. The method was tested using Monte Carlo simulation. Each Monte Carlo experiment simulated 1,000 confusion matrices, for sample sizes of 50 and 100 cases, at fixed population values of kappa. The results showed a slight systematic error with decreasing values of kappa. This result may be explained because the experiment was limited by the discrete nature of the simulation as discussed in section 3.6.2. The accuracy of the confidence interval is thus uncertain.

4.6 Summary

This chapter has given a brief overview of methods for the statistical analysis of confusion matrices. The simplest method is to count the proportion of cases on the diagonal, however this takes no account of the number that may occur by chance alone. The chi–squared statistic could be used, but this only provides a measure of association, not agreement. However chi–squared is an ideal statistic for the Monte Carlo simulations used to test the new methods proposed in this thesis.

In order to measure chance corrected agreement Cohen proposed the kappa statistic which measures agreement on a continuous scale of –1 to +1. Linguistic categories were introduced to ease the interpretation of these kappa values. There is also a weighted kappa statistic that can be used with weighted confusion matrices.

The kappa and weighted kappa statistics have Gaussian distributions for large sample sizes and therefore a 95% confidence interval can be calculated. Different versions of the kappa confidence interval have been tested by Monte Carlo simulations, and by direct computation, for both the general case, and the 2×2 case. Due to theoretical limitations, these experiments provide little information at small sample size, and therefore it is not known at what sample size, and kappa value, the Gaussian becomes an acceptable approximation for the distribution of kappa.

5 New Methods for Generating pdfs

This chapter presents novel mathematical and computational methods for calculating the pdf of the various statistical measurements of ROC curves and confusion matrices.

All these methods are based on very simple equations of the form:

$$p_n \propto \prod_{i=0}^{n} x_i^{a_i} \qquad \text{where} \qquad x_i \geq 0 \qquad \text{and} \qquad \sum_{i=0}^{n} x_i = 1 \qquad (5.1)$$

If n is 3, equation 5.1 computes a probability within a cube with the axes x_0, x_1, x_2, which is a Dirichlet distribution. Because of the constraint that $x_0 + x_1 + x_2 = 1$, the probability is actually constrained to a plane across the diagonal of the cube. In general, the equation is constrained to a $n - 1$ dimensional hyperplane in n dimensional space. Summing, or integrating, the probability across all n dimensional points on the plane that have the same property of interest allows the pdf of the property to be constructed. This is fine in theory, but producing anything useful requires mathematical and computational techniques to extract it. This chapter presents these techniques for calculating pdf, and hence the posterior interval, of:

- a single point nonparametric ROC curve;
- each point of a multi-point nonparametric ROC curve;
- the nonparametric AUC;
- the parametric AUC, and the parametric parameters;
- the weight of a weighted confusion matrix.

The first two are analytical methods, the rest are based on algorithms.

This work starts from the same point as Hilgers ([42], section 3.2), but goes further.

All the methods were implemented in software, and tested by Monte Carlo simulations. Chapter 6 documents the Monte Carlo tests, and Chapter 7 illustrates the methods using examples from the literature.

5.1 Single Nonparametric ROC Point

Calculating the posterior interval for a single point ROC 'curve' is based upon asking the following question for every possible point on the surface of the ROC graph:

'If this point represents the true Hit Rate and False Alarm Rate of the population, what would be the probability of actually obtaining the sample?'

If that question can be answered for every point on the graph, and normalised so that the total probability of every point on the surface sums to 1, a probability density function for the true Hit Rate and False Alarm Rate can be generated. By dividing the surface into a fine grid, and integrating the expression for the probability of every point over each square of the grid, the surface can then be presented as a 3D mesh, or contour lines can be drawn to enclose an arbitrary percentage of the probability, e.g. 95% of the probability, which gives the 95% posterior interval for the location of the true Hit Rate and False Alarm Rate.

Consider the full situation where:

y is the Hit Rate of the population, given as a probability;

x is the False Alarm Rate of the population, given as a probability;

f is the frequency of disease events in the population, given as a probability;

b_0 is the number of true positives in the sample;

a_0 is the number of false positives in the sample;

b_1 is the number of false negatives in the sample;

a_1 is the number of true negatives in the sample.

Then, P, the probability of a ROC point being at the location (x, y), is given by the product of three terms. The first term, is the probability of obtaining b_0+b_1 diseased cases in $a_0+a_1+b_0+b_1$ cases when the probability of disease is f. The second term, is the probability of obtaining a_0 False Alarms in a_0+a_1 healthy cases when the probability of a False Alarm is x; and the third term, is the probability of obtaining b_0 Hits in b_0+b_1 diseased cases when the probability of a Hit is y :

$$P_{xyfb_0a_0b_1a_1} = \binom{a_0 + a_1 + b_0 + b_1}{b_0 + b_1} f^{b_0+b_1} (1-f)^{a_0+a_1} \binom{a_0 + a_1}{a_0} x^{a_0} (1-x)^{a_1} \binom{b_0 + b_1}{b_0} y^{b_0} (1-y)^{b_1}$$

In order to normalise the probability at each point (x, y), to sum to 1.0 when integrated across the whole surface, the probability is divided by the integral over the surface:

$$PointProbability_{xy} =$$

$$\frac{\binom{a_0 + a_1 + b_0 + b_1}{b_0 + b_1} f^{b_0+b_1} (1-f)^{a_0+a_1} \binom{a_0 + a_1}{a_0} x^{a_0} (1-x)^{a_1} \binom{b_0 + b_1}{b_0} y^{b_0} (1-y)^{b_1}}{\binom{a_0 + a_1 + b_0 + b_1}{b_0 + b_1} f^{b_0+b_1} (1-f)^{a_0+a_1} \binom{a_0 + a_1}{a_0} \int_0^1 x^{a_0} (1-x)^{a_1}\, dx \binom{b_0 + b_1}{b_0} \int_0^1 y^{b_0} (1-y)^{b_1}\, dy}$$

$$= \frac{x^{a_0} (1-x)^{a_1}}{\int_0^1 x^{a_0} (1-x)^{a_1}\, dx} \quad \frac{y^{b_0} (1-y)^{b_1}}{\int_0^1 y^{b_0} (1-y)^{b_1}\, dy} \tag{5.2}$$

The *Beta* function can be used to substitute for the integrals. The *Beta* function, by definition [84][85], and given that m and n are integers, is:

$$\beta(m,n) = \int_0^1 x^{m-1} (1-x)^{n-1}\, dx = \frac{(m-1)!(n-1)!}{(m+n-1)!}$$

If $m = a_0 + 1$ and $n = a_1 + 1$

$$\int_0^1 x^{a_0+1-1} (1-x)^{a_1+1-1}\, dx = \frac{(a_0 + 1 - 1)!(a_1 + 1 - 1)!}{(a_0 + 1 + a_1 + 1 - 1)!}$$

$$\therefore \int_0^1 x^{a_0} (1-x)^{a_1}\, dx = \frac{a_0!\ a_1!}{(a_0 + a_1 + 1)!} \tag{5.3}$$

Substitute (5.3) into (5.2) to give:

$$PointProbability_{xy} = \frac{x^{a_0} (1-x)^{a_1}}{\frac{a_0! a_1!}{(a_0+a_1+1)!}} \quad \frac{y^{b_0} (1-y)^{b_1}}{\frac{b_0! b_1!}{(b_0+b_1+1)!}}$$

To represent the surface, it is divided into a fine grid and the probability of each quantized grid square is calculated by integrating the probability at a point, over the area of each grid square. The integral over the area is equal to the product of two one diminsional

integrals along the Hit Rate and False Alarm Rate axes. Therefore two column vectors, $\boldsymbol{X}$ and $\boldsymbol{Y}$, each with i elements, are defined to hold the one dimensional integrals:

$$\boldsymbol{X}_i = \frac{\int_{\frac{i-1}{n}}^{\frac{i}{n}} x^{a_0} (1-x)^{a_1} \, dx}{\frac{a_0! a_1!}{(a_0 + a_1 + 1)}} \qquad \text{and} \qquad (5.4)$$

$$\boldsymbol{Y}_i = \frac{\int_{\frac{i-1}{n}}^{\frac{i}{n}} y^{b_0} (1-y)^{b_1} \, dy}{\frac{b_0! b_1!}{(b_0 + b_1 + 1)}} \qquad \text{For all } i \text{ from } i\text{=}1 \text{ to } i\text{=}n \qquad (5.5)$$

The joint probability density function, quantized as a fine grid, is therefore the product of the two column vectors:

$$Surface = \boldsymbol{X} \cdot \boldsymbol{Y}^T \qquad (5.6)$$

The numerators of the vectors $\boldsymbol{X}$ and $\boldsymbol{Y}$ ((5.4), (5.5)) are given in terms of an expression of the following form:

$$Numerator = \int_q^r x^{a_0} (1-x)^{a_1} \, dx \qquad (5.7)$$

Where q is the probability at the lower boundary of the element, and r is the probability at the upper boundary of the element.

$$\therefore \; Numerator = \int_0^r x^{a_0} (1-x)^{a_1} \, dx \; - \int_0^q x^{a_0} (1-x)^{a_1} \, dx \qquad (5.8)$$

Dealing with one partial beta function at a time, where either q or r can be substituted for s:

$$\int_0^s x^{a_0} (1-x)^{a_1} \, dx \; = \int_0^s x^{a_0} \left(\, (1-s) + (s-x) \, \right)^{a_1} \, dx$$

Applying the binomial expansion:

$$= \int_0^s x^{a_0} \sum_{k=0}^{a_1} \frac{a_1!}{k!(a_1-k)!} (1-s)^k (s-x)^{a_1-k} \, dx$$

$$= \sum_{k=0}^{a_1} \frac{a_1!}{k!(a_1-k)!} (1-s)^k \int_0^s x^{a_0} (s-x)^{a_1-k} \, dx \tag{5.9}$$

Change the limits on the integral from *0* to *s*, to *0* to *1*, by letting $x = s\ t$, which implies $dx = s\ dt$, and letting $a_2 = a_1 - k$:

$$\int_0^s x^{a_0} (s-x)^{a_2} \, dx = \int_0^1 (s\ t)^{a_0} (s - s\ t)^{a_2} \ s \ dt$$

$$= \int_0^1 s^{a_0} \ t^{a_0} \ (s\ (1-t))^{a_2} \ s \ dt$$

$$= \int_0^1 s^{a_0} \ s^{a_2} \ s \ t^{a_0} \ (1-t)^{a_2} \ dt$$

$$= s^{a_0+a_2+1} \int_0^1 t^{a_0} \ (1-t)^{a_2} \ dt$$

(appendix section A.1 gives this derivation in seven steps)

Substituting the modified Beta function as given by (5.3):

$$= s^{(a_0+a_2+1)} \ \frac{a_0! \ a_2!}{(a_0+a_1+1)!} \tag{5.10}$$

Substituting (5.10) into (5.9):

$$\int_0^s x^{a_0} \ (1-x)^{a_1} \ dx \ = \sum_{k=0}^{a_1} \frac{a_1!}{k!(a_1-k)!} \ (1-s)^k \ s^{a_0+a_1-k+1} \ \frac{a_0! \ (a_1-k)!}{(a_0+a_1-k+1)!}$$

$$= a_0!a_1! \sum_{k=0}^{a_1} \frac{(1-s)^k \; s^{a_0+a_1-k+1}}{k! \; (a_0+a_1-k+1)!} \tag{5.11}$$

Substituting (5.11) into (5.8) and then substituting into (5.7) and simplifying:

$$Numerator = a_0!a_1! \sum_{k=0}^{a_1} \frac{r^{a_0+a_1+1-k} \; (1-r)^k - q^{a_0+a_1+1-k} \; (1-q)^k}{k!(a_0+a_1+1-k)!} \tag{5.12}$$

Substituting (5.12) into (5.4) gives the expression for each element of the $\boldsymbol{X}$ vector:

$$\boldsymbol{X}_i = \frac{a_0!a_1!}{\frac{a_0!a_1!}{(a_0+a_1+1)!}} \sum_{k=0}^{a_1} \frac{\left(\frac{i}{n}\right)^{a_0+a_1+1-k} \left(1-\frac{i}{n}\right)^k - \left(\frac{i-1}{n}\right)^{a_0+a_1+1-k} \left(1-\frac{i-1}{n}\right)^k}{k!(a_0+a_1+1-k)!}$$

For all i from i=1 to i=n

Which simplifies to:

$$\boldsymbol{X}_i = (a_0+a_1+1)! \sum_{k=0}^{a_1} \frac{\left(\frac{i}{n}\right)^{a_0+a_1+1-k} \left(1-\frac{i}{n}\right)^k - \left(\frac{i-1}{n}\right)^{a_0+a_1+1-k} \left(1-\frac{i-1}{n}\right)^k}{k!(a_0+a_1+1-k)!}$$

For all i from i=1 to i=n (5.13)

Similarly, for $\boldsymbol{Y}$ (by substituting (5.12) into (5.5) and simplifying):

$$\boldsymbol{Y}_i = (b_0+b_1+1)! \sum_{k=0}^{b_1} \frac{\left(\frac{i}{n}\right)^{b_0+b_1+1-k} \left(1-\frac{i}{n}\right)^k - \left(\frac{i-1}{n}\right)^{b_0+b_1+1-k} \left(1-\frac{i-1}{n}\right)^k}{k!(b_0+b_1+1-k)!}$$

For all i from i=1 to i=n (5.14)

Diagrammatically, this can be seen in Figure 5.1, which shows a sketch of a pdf generated by integrating the probability that the population has a False Alarm Rate of x, and a Hit Rate of y, across the surface.

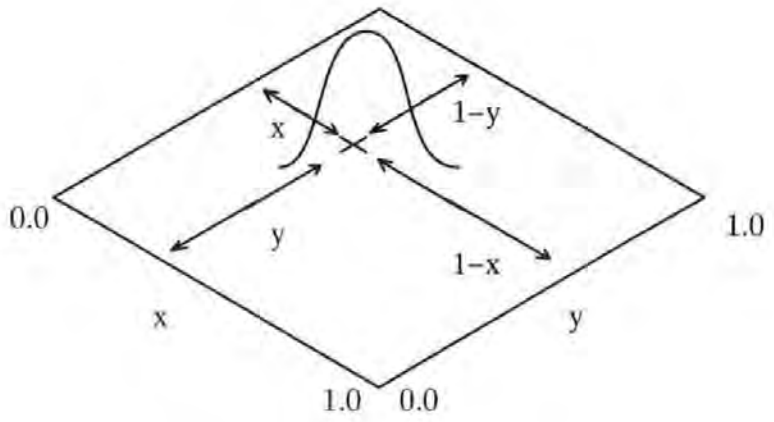


Figure 5.1 Pdf of a single ROC point

In order to later expand the analysis to the multi-point case it is helpful at this stage to let: $x_0 = x$, $x_1 = 1 - x$, $y_0 = y$ and $y_1 = 1 - y$.

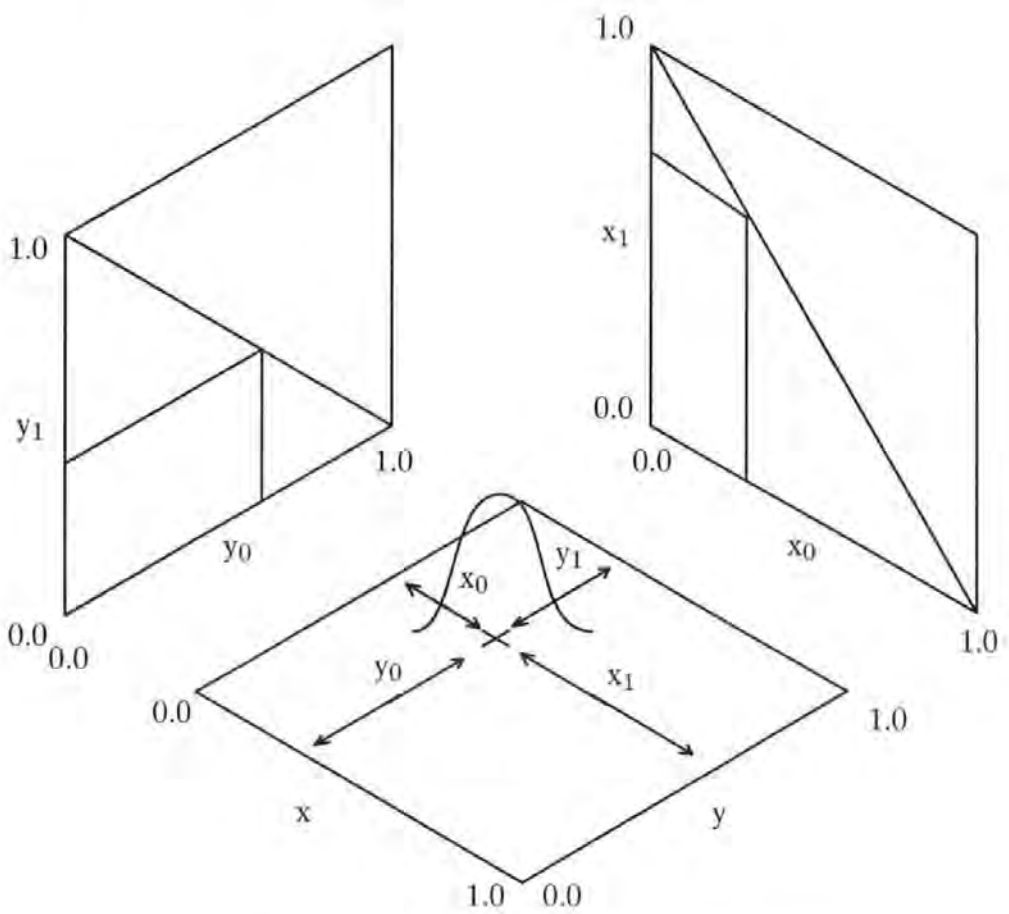


Figure 5.2 Single ROC point from two linear pdfs

From equation (5.6), the pdf is the product of two components, the x component and the y component. Each of these components can be plotted as graphs of x_0 against x_1, and

y_0 against y_1 respectively (Figure 5.2). In this case the relationship between x_0 and x_1, and y_0 and y_1 constrains each component pdf to a line on the respective plane.

5.2 Multiple Nonparametric ROC Points

The analysis can now be expanded to the general case of multiple ROC points. Since it has been shown above, that the surface can be treated as a product of two probability density vectors, one for Hit Rate and the other for False Alarm Rate, this discussion will examine only one vector, the *X'*, or False Alarm Rate vector, the identical method being applicable to the *Y'*, or Hit Rate Vector.

For a ROC curve of n points, there are $n+1$ classifications of events (threshold ranges). Let there be a_i occurrences of event e_i, where $i = 0, ..., n$ (see Table 1.2). Let the true probability of event e_i be x_i. Now, an extension of the hypothesis stated above can be applied, by asking the following question, for every point:

'If this point, $x_0, x_1, ..., x_n$, represents the true probability of events, $e_0, e_1, ..., e_n$, in the population, what would be the probability of actually obtaining the sample $a_0, a_1, ..., a_n$?'

If that question can be answered for every point, and normalised such that the total probability of every point in the hyper–volume sums to 1, the probability density function in $n+1$ dimensional space could be calculated.

By multinomial from the numerator of equation (5.2) for the single ROC point case, the probability P', of a point lying on a hyperplane in $n+1$ dimensional space can be written as:

$$P' \propto \left[\prod_{i=0}^{n-1} x_i^{a_i}\right]\left[1 - \sum_{i=0}^{n-1} x_i\right]^{a_n} \qquad \text{where} \quad x_i \geq 0 \quad \text{and} \quad \sum_{i=0}^{n} x_i = 1$$

To represent the probability density function on a two dimensional ROC graph, the $n+1$ dimensional probability density function must be mapped into one dimension, i.e. to a *X'* vector for the False Alarm Rate, or a *Y'* vector for the Hit Rate, and the two dimensional ROC surface formed as the product of the *X'* and *Y'* vectors.

Note that each point on the ROC curve represents a different combination of events. The first point represents e_0 events only, but the second point represents the e_0 plus the e_1 events, the third point e_0, e_1 plus e_2 events, and so on. This adds a subtlety to the way the hyperplane is mapped to the linear probability density function for each point.

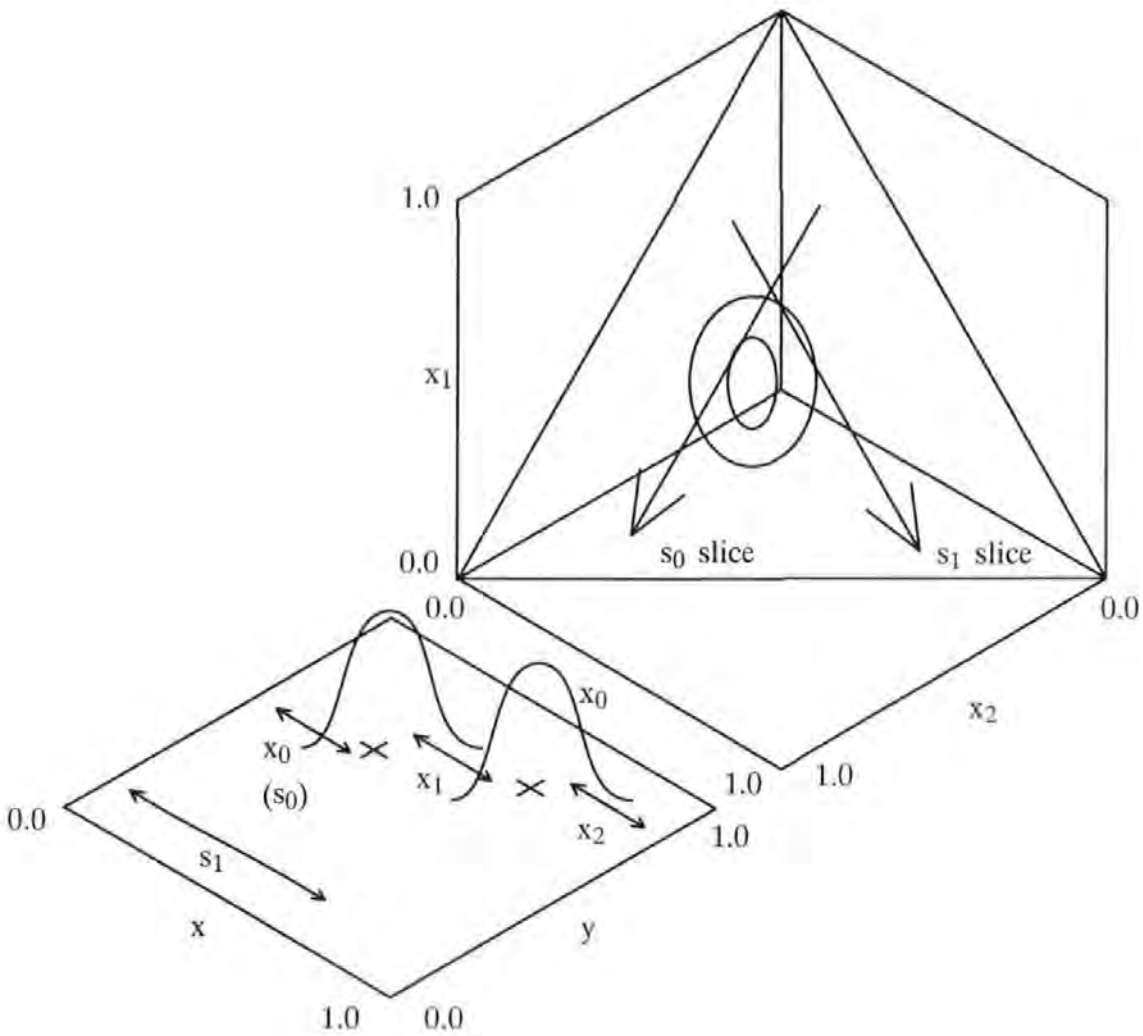


Figure 5.3 Two point ROC pdf

Diagrammatically, this is illustrated for the two point ROC curve in Figure 5.3. The pdf of the x component (drawn as two contours) lies on a plane within a cube. The plane satisfies the constraint $x_0+x_1+x_2 = 1.0$. The pdf is a Dirichlet distribution.

Let s_0 be the actual probability of the first ROC point. The first point represents only the true probability of the e_0 events. As s_0 varies from 0 to 1, it is directly related to x_0 by the relationship $s_0 = x_0$. The plane is mapped into the line by integrating the pdf across

slices at right angles to the x_0 axis, using a Dirichlet integral, as shown by the the s_0 slice arrow.

Let s_1 be the actual probability of the second ROC point. The second point is the combined probability of the e_0 events, plus e_1 events, in the population. If does not matter what the individual probability of e_0 events is, or what the individual probability of e_1 events is, only the combined probability matters. As s_1 varies from 0 to 1, x_0 and x_1 are constrained by the relation $s_1 = x_0 + x_1$. The plane is mapped into the line by integrating across slices at right angles to the x_0, x_1 plane, or at right angles to the line $x_0 + x_1 + x_2 = 1$, as shown by the s_1 slice arrow.

The same analysis applies to ROC curves with more than two points. If there are three points the pdf lies on a three dimensional hyperplane in a four dimensional hypercube.

Let s_2 be the actual probability of the third ROC point. The third point is the combined probabilities of e_0, e_1 and e_2 events. As s_2 varies from 0 to 1, x_0, x_1 and x_2 are constrained by the relationship $s_2 = x_0 + x_1 + x_2$. The hyperplane is mapped into the line by integrating across the three dimensional diagonal slices at right angles to the line $x_0 + x_1 + x_2 = 1$.

By way of example, the integrals for a four point ROC curve are given below.

5.2.1 Four ROC points, 1st Point

$$f(s_0) = \int_0^{1-s_0} \int_0^{1-s_0-x_1} \int_0^{1-s_0-x_1-x_2} {s_0}^{a_0} {x_1}^{a_1} {x_2}^{a_2} {x_3}^{a_3} (1 - s_0 - x_1 - x_2 - x_3)^{a_4} \, dx_3 dx_2 dx_1 \tag{5.15}$$

The variable s_0 ranges from 0 to 1 across the probability density function. In this case s_0 is equivalent to x_0. Since the function is constrained to the hyperplane $x_0+x_1+x_2+x_3+x_4 = 1$, x_1 is thus confined to the range $1-s_0$, which are therefore the limits of the outer integral. Similarly, x_2 is then confined to the range 0 to $1-s_0-x_1$, the limits of the middle integral, and x_3 is confined to the range 0 to $1-s_0-x_1-x_2$, the limits of the inner integral. The expression is kept on the hyperplane by substituting $x_4 = 1 - s_0 - x_1 - x_2 - x_3$, in the last term.

Substituting (5.15) into (5.10), above, where s is, in turn, $1-s_0-x_1-x_2$, $1-s_0-x_1$, and $1-s_0$:

$$f(s_0) = \int_0^{1-s_0} \int_0^{1-s_0-x_1} {s_0}^{a_0}\ {x_1}^{a_1}\ {x_2}^{a_2}\ (1 - s_0 - x_1 - x_2)^{a_3+a_4+1}\ \frac{a_3!\ a_4!}{(a_3 + a_4 + 1)!}\ dx_2 dx_1$$

$$= \int_0^{1-s_0} {s_0}^{a_0}\ {x_1}^{a_1}\ (1 - s_0 - x_1)^{a_2+a_3+a_4+2}\ \frac{a_2!\ (a_3 + a_4 + 1)!}{(a_2 + a_3 + a_4 + 2)!}\ \frac{a_3!\ a_4!}{(a_3 + a_4 + 1)!}\ dx_1$$

$$= {s_0}^{a_0}\ (1 - s_0)^{a_1+a_2+a_3+a_4+3}\ \frac{a_1!\ (a_2 + a_3 + a_4 + 2)!}{(a_1 + a_2 + a_3 + a_4 + 3)!}\ \frac{a_2!\ a_3!\ a_4!}{(a_2 + a_3 + a_4 + 2)!}$$

$$= {s_0}^{a_0}\ (1 - s_0)^{a_1+a_2+a_3+a_4+3}\ \frac{a_1!\ a_2!\ a_3!\ a_4!}{(a_1 + a_2 + a_3 + a_4 + 3)!}$$

5.2.2 Four ROC points, 2nd point

$$f(s_1) = \int_0^{s_1} \int_0^{1-s_1} \int_0^{1-s_1-x_2} {x_0}^{a_0}\ (s_1 - x_0)^{a_1}\ {x_2}^{a_2}\ {x_3}^{a_3}\ (1 - s_1 - x_2 - x_3)^{a_4}\ dx_3 dx_2 dx_0$$

Again, the variable s_1 ranges from 0 to 1 across the probability density function. In this case $s_1 = x_0 + x_1$, and therefore x_0 is confined to the range 0 to s_1, which are therefore the limits of the outer integral. In the second term $s_1 - x_0$ is substituted for x_1, and, in the last term, s_1 is substituted for $-(x_0+x_1)$. The range of x_2 is then confined to the range 0 to $1-s_1$, the limits of the middle integral, and x_3 is confined to the range 0 to $1-s_1-x_1$, the limits of the inner integral.

$$f(s_1) = {s_1}^{a_0+a_1+1}\ \frac{a_0!\ a_1!}{(a_0 + a_1 + 1)!}\ (1 - s_1)^{a_2+a_3+a_4+2}\ \frac{a_2!\ a_3!\ a_4!}{(a_2 + a_3 + a_4 + 2)!}$$

5.2.3 Four ROC points, 3rd point

$$f(s_2) = \int_0^{s_2} \int_0^{s_2-x_0} \int_0^{1-s_2} {x_0}^{a_0}\ {x_1}^{a_1}\ (s_2 - x_0 - x_1)^{a_2}\ {x_3}^{a_3}\ (1 - s_2 - x_3)^{a_4}\ dx_3 dx_1 dx_0$$

Here, $s_2 = x_0 + x_1 + x_2$. This is used in the third and fifth term. The outer and middle integrals are limited by this expression. The inner integral is constrained by the hyperplane.

$$f(s_2) = s_2^{a_0+a_1+a_2+2} \frac{a_0!\ a_1!\ a_2!}{(a_0 + a_1 + a_2 + 2)!} (1 - s_2)^{a_3+a_4+1} \frac{a_3!\ a_4!}{(a_3 + a_4 + 1)!}$$

5.2.4 Four ROC points, 4th point

$$f(s_3) = \int_0^{s_3} \int_0^{s_3-x_0} \int_0^{s_3-x_0-x_1} x_0^{a_0}\, x_1^{a_1}\, x_2^{a_2}\, (s_3 - x_0 - x_1 - x_2)^{a_3}\, (1 - s_3)^{a_4}\, dx_2 dx_1 dx_0$$

Here, $s_3 = x_0 + x_1 + x_2 + x_3$. This is used in the fourth and fifth term. All the integrals are limited by this expression. The hyperplane constraint is only evident in the last term.

$$f(s_3) = s_3^{a_0+a_1+a_2+a_3+3} \frac{a_0!\ a_1!\ a_2!\ a_3!}{(a_0 + a_1 + a_2 + a_3 + 3)!} (1 - s_3)^{a_4}$$

Appendix section A.2 gives the raw expressions for each point of ROC curves with between one and four points (10 expressions). Appendix section A.3 expands on the above by giving all the intermediate steps in integrating the expressions for each point of a four point ROC curve.

5.2.5 The General Result for Multiple ROC Points

When any $f(s_n)$ is normalised by dividing it by its integral, as shown for the one ROC point example in equation (5.2), the factorial terms cancel out. Integration of the expressions for each point of one, two, three, and four point ROC curves reveals a pattern, which by induction generalises to:

$$f(s) = s^{\sum_{i=0}^{n-1} a_i + n - 1} (1 - s)^{\sum_{j=n}^{m} a_j + m - n} \tag{5.16}$$

Where n is the number of the point, and m the total number of points in the curve. Thus the expression for the pdf projected onto one dimension includes the terms n and m which can be regarded as *'order terms'*.

If:

$$a'_0 = \sum_{i=0}^{n-1} a_i + n - 1 \qquad \text{and} \qquad a'_1 = \sum_{j=n}^{m} a_j + m - n$$

The multi-point ROC equations are in exactly the same form as the numerator of the single point ROC equation (5.2) and since the denominator is the intergal over the whole hyper-volume used to normalise the distribution to sum to 1.0, the same method can be applied to calculate the probability density function.

It should be noted that the probability density function of n ROC points actually exists in $2n$ dimensional space. The mapping to n 2D probability surfaces, overlaid on one ROC curve, is merely a convenient representation of this single multidimensional probability density function.

5.2.6 Order Terms

The *'order terms'* in equation 5.16 have the consequence that the number of classification categories effects the posterior intervals of the ROC points. This is a consequence of the uniform Bayesian prior used to derive the equations, which is effectively the function:

$$UniformBayesianPrior = x^0\ (1-x)^0$$

Using this uniform Bayesian prior distribution, the posterior probability of a population point is:

$$p_x = x^{a_0}\ (1-x)^{a_1}$$

However, the order terms will disappear if an uninformative prior [15] is used:

$$UninformativeBayesianPrior = x^{-1}\ (1-x)^{-1}$$

Using this uninformative prior gives a posterior probability of a population point as:

$$p_x = x^{a_0-1}\ (1-x)^{a_1-1}$$

However, if $a_0 = 0$ or $a_1 = 0$ the integral of this function is infinity, which is an improper posterior distribution:

$$\int_0^1 x^{a_0-1}\ (1-x)^{a_1-1}\ dx = \infty$$

The Monte Carlo simulations (Chapter 6) of each of the novel methods presented in this thesis use a uniform prior distribution as a pragmatic way of proving each method works. However, this does not preclude using an uninformative Bayesian prior with any of the methods for particular applications and data sets.

5.2.7 Software Implementation

A computer program was written in C++, to perform the calculation, and plot the 95% posterior interval contour. The actual code optimised the mathematical expression (5.13) (and (5.14)) by only calculating, and storing in a vector, the boundary values:

$$BoundaryValue = (a_0 + a_1 + 1)! \sum_{k=0}^{a_1} \frac{\left(\frac{i}{n}\right)^{a_0+a_1+1-k} \left(1 - \frac{i}{n}\right)^k}{k!(a_0 + a_1 + 1 - k)!}$$

Each element was then calculated as the difference between two boundary values.

Many terms in the expression were pre-calculated and accessed from look-up tables.

The surface was then calculated by a product of the two vectors. The 'tiles' of the surface were then sorted by probability, largest first, and marked in order, from the largest, as being inside the posterior boundary until the sum of the marked 'tiles' equalled $z\%$ of the sum of all the tiles. A boundary drawing algorithm was then applied, to draw around the marked area to give the $z\%$ posterior boundary.

The C++ source code for generating the pdf is given in appendix section C.1. The code runs in time proportional to the number of elements in the pdf multiplied by the lesser of the two powers, a_0 and a_1. The program produced an encapsulated postscript file of a plot of the 95% posterior boundary of the points of up to two ROC curves.

The number of 'tiles' along each side of the surface is an exact power of two. The results presented in this thesis use a surface of 512 by 512 'tiles', but any power of two could be used. The False Alarm Rate is divided into 512 thin slices where the probability is integrated over the range 0 to 1/512, 1/512 to 2/512, ..., 511/512 to 512/512. The Hit Rate is sliced similarly. This contrasts with other pdf calculations in this chapter, for instance the pdf of the nonparametric AUC, where the pdf is sampled at evenly spaced points.

5.2.7.1 Extending the Exponent Range

Because of the range of the exponent required in the calculation it was necessary to use the operator overloading facilities of C++ and write a class for a floating point number with additional exponent range. In this REAL precision class, the mantissa was represented by a double precision variable and the exponent by a long integer variable.

Testing revealed that using the class was only necessary when the power was more than 128. Because calculations using the defined REAL variables took about three times as long as calculations using double precision variables, and because it was estimated some Monte Carlo simulations would take months, all the software was written as C++ templates so that the compiler would automatically produce REAL precision and double precision executable versions from the same source code and so time could be saved in simulations.

All the graphics software for plotting posterior intervals uses REAL precision whatever the number of cases, while all the Monte Carlo simulation software uses double precision for sample sizes up to 128 and REAL precision for sample sizes above 128.

5.2.7.2 Power Table

Calculating powers of numbers between 0 and 1 is a ubiquitous operation in the software that is computationally expensive, so it was optimized by using a look–up table of pre–computed values. All the grid sizes of probability density functions were chosen to give an exact binary fraction to the mantissa of these power calculations in order to minimise rounding errors. Raising zero to the power of zero gives a value that is implementation dependent, but here a value of 1.0 was used.

Before examining zero to the power of zero, the value of 1! needs to be defined. If a coin if flipped once and gives a sequence of one head and zero tails there is one unique sequence, therefore:

$$\frac{1!}{1!\ 0!} = 1$$

$$\therefore 0! = 1$$

If a very biased coin, where the probability of a head is 1.0, is flipped once, the probability of a head must be 1.0:

$$\begin{bmatrix} a_0 + a_1 \\ a_0 \end{bmatrix} x^{a_0} (1 - x)^{a_1} = 1$$

$$\therefore \frac{1!}{1!\ 0!}\ 1^1\ 0^0 = 1$$

$$\therefore 1 \times 1 \times 0^0 = 1$$

$$\therefore 0^0 = 1$$

In other areas of mathematics this might be a moot point, but here it will be taken as an axiom.

5.2.7.3 Graphics Library

The software produces graphs as encapsulated postscript files using a simple, short, bespoke graphics library. This had the advantage of allowing the entire suite of programs to be written in portable C++ while producing graphical output that can be displayed by many software packages on different hardware platforms, including the word processor used to write this thesis.

5.3 Comparing Nonparametric ROC Points

The nonparametric ROC points from two systems, A and B, can be objectively compared from the pdf of the ROC points. The method presented here assumes the two ROC curves are uncorrelated, which would occur if each system was tested with different samples. Correlated ROC curves are discussed in section 8.3.7.

The pdf of each ROC points of each system can be calculated as in section 5.2. The pdf of the first point for system A can then be compared with the first point for system B, and so on for all pairs of points. By definition, each grid square of the pdf of the n^{th} point of system A gives the probability that the population False Alarm Rate and Hit Rate of that system are in that grid square. Similarly for the pdf of the n^{th} point of system B. Therefore, if just the n^{th} point is considered, the combined probability that system A actually has a False Alarm Rate of i and a Hit Rate of j, and that system B actually has a False Alarm Rate of $i+k$ and Hit Rate of $j+l$ can be given as:

$$JointProbability = Xa_i \cdot Ya_j \cdot Xb_{i+k} \cdot Yb_{j+l}$$

Where Xa_i is the i^{th} element of the $\boldsymbol{X}$ vector of system A as defined in (5.13);

Ya_j is the j^{th} element of the $\boldsymbol{Y}$ vector of system A as defined in (5.14);

Xb_{i+k} is the $i+k^{th}$ element of the $\boldsymbol{X}$ vector of system B as defined in (5.13);

Yb_{j+l} is the $j+l^{th}$ element of the $\boldsymbol{Y}$ vector of system B as defined in (5.14).

The difference in population False Alarm Rate between systems A and B is therefore given by k, and the difference in Hit Rate by l. The joint probability thus gives the probability of a particular difference in False Alarm Rate and Hit Rate, for one particular False Alarm Rate and Hit Rate, of one system (the reference system). The total probability for the difference can be obtained by summing over all possible False Alarm and Hit Rates of the reference system:

$$P_{k,l} = \sum_{i=1}^{i=n} \sum_{j=1}^{j=n} Xa_i \cdot Ya_j \cdot Xb_{i+k} \cdot Yb_{j+l}$$

Which is computationally easier to calculate as:

$$P_{k,l} = \sum_{i=1}^{i=n} Xa_i \cdot Xb_{i+k} \cdot \sum_{j=1}^{j=n} Ya_j \cdot Yb_{j+l} \qquad (5.17)$$

It is possible for the difference in False Alarm Rate, k, to vary from $-n+1$ to $n-1$, similarly with the Hit Rate, l. System B's vectors will thus be out of bounds in some cases. In these conditions the probability is zero because the situation is physically impossible.

5.3.1 Software Implementation

Two computer program were written in C++, to perform the calculations, and plot the 95% posterior intervals. The code to calculate the pdf is given in appendix section C.2. The first program generates one graph for each pair of points on two ROC curves. Each graph shows the posterior interval of the difference between a pair of points. This allows detailed information about each pair to be shown. The second program plots all these posterior intervals on one ROC graph. Each posterior interval is located midway between the location of its parent points. This provides a better overview, but allows less detail to be presented.

5.3.1.1 Detailed Graph

The first program plots graphs of the difference in Hit Rate on a scale of −1.0 to +1.0, and of difference of False Alarm Rate on a scale of −1.0 to +1.0. Figures 7.7 and 7.8 show examples of this type of graph. Each graph is divided into four quadrants along the zero lines. The upper left quadrant represents the situation where a system has both a higher Hit Rate, and a lower False Alarm Rate than another. This is usually considered highly desirable behaviour. The lower left quadrant represents the situation where a system has a lower False Alarm Rate, but lower Hit Rate than another, and the upper right quadrant where a system has a higher Hit Rate but a higher False Alarm Rate than another. This might be what is sought in certain circumstances. The lower right quardant represents systems that have a lower Hit Rate and a higher False Alarm Rate than another. This is usually a very undesirable situation.

To help quantify the difference between systems, the probability that the actual difference is in each of the four quadrants is calculated by summing the pdf over each quadrant and printing the sums in the corners of each graph. The only slight difficulty with this is that the quantized grid has finite thickness zero lines which therefore have a finite probability. This probability is partitioned between the quadrants. The 0, 0 'tile' is equally partitioned into the four quadrants. The other 'tiles' are equally partitioned into their two bounding quadrants.

Note, that this difference operator is not commutive. However, the difference between system A and B can be transformed into the difference between B and A by double reflection along the x and y axis of the graph.

5.3.1.2 Combined Graph

The second program generates exactly the same posterior intervals, but plots them against the background of a ROC curve. Figures 7.5 and 7.6 give examples of this type of graph. Each point is given a scale in the form of cross hairs with a size difference of ± 0.2. The 0, 0 point of the cross hairs is located at the mean of the pdfs of the parent points. Thus it is easy to relate the graph of differences to a graph showing both sets of parent ROC

points. The quadrant information is not plotted on the graph, as this would be confusing, instead the program produces a text file giving this information.

5.4 Nonparametric AUC

As has been shown in the previous section that the probability density function of an n point nonparametric ROC curve actually exists in $2n$ dimensional space, or more precisely as a $2n$ dimensional hyperplane in a $2(n+1)$ dimensional hypercube. Given a set of pairs of x and y values for a ROC curve, the Area Under the Curve can be calculated by using the trapezoid rule (section 3.4.1). The AUC for every point on the hyperplane can thus be calculated. If the probability of every point with the same AUC can be integrated across the hyperplane, the pdf of the AUC can be generated.

The proposed algorithm is derived from an algorithm to find the shortest path between two nodes on a general graph, and variations of it have been used by the author to provide an 'exact' (quantized) solution to Fisher's Two Tail statistic [87]. Of all the computational solutions proposed in this thesis, this is the least efficient, running in time proportional to the fifth power of the quantized grid size, but this is still practical on a fast modern PC.

Consider the graph in Figure 5.4, and the problem of finding the shortest path from A to H.

The naive solution is to perform a breadth first search starting from node A, and to compare the length of all paths from A to H. The search would proceed as follows (the accumulating path length is given in parentheses):

A → B (1.0)

A → C (1.0)

A → D (2.3)

A → E (2.2)

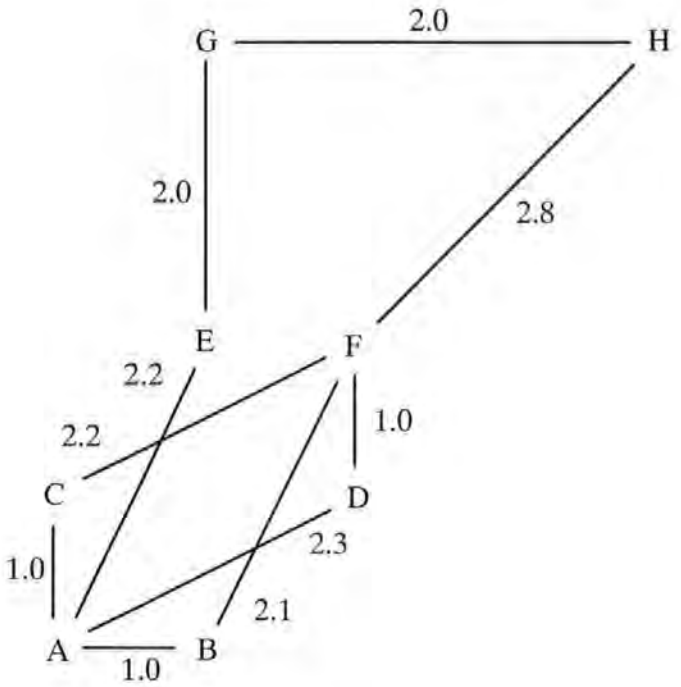


Figure 5.4 Find the shortest path from A to H

A → B → F (3.1)

A → C → F (3.2)

A → D → F (3.3)

A → E → G (4.2)

A → B → F → H (5.9)

A → C → F → H (6.0)

A → D → F → H (6.1)

A → E → G → H (6.2)

Since there are no other paths, the algorithm terminates here, and the shortest path from A to H (with a length of 5.9) is A→B→F→H.

Note however that there are three paths through node F, and that if the solution includes node F, it must contain the shortest path from A to F. Thus there is no point in keeping all the paths from A to F in the search space. All except the shortest can be removed.

Now the same idea can be applied to produce an algorithm for finding the pdf of the AUC of a nonparametric ROC curve.

Assume the ROC graph is quantized into 5 by 5 points as shown in Figure 5.5, and the ROC curve has two points. The first point can be anywhere on the grid, therefore there are 25 path segments from the origin (node A, at location 0,0) to each of these points. (This includes the possibility that the first segment is zero length and goes from 0,0 to 0,0). For each of these 25 first points, the second point can be anywhere above or to the right of its respective first point, including zero distance. For example, if the first point is at 0.5, 0.5, the second point can be in 9 possible locations (including 0.5, 0.5). This gives 225 possible paths in total. From the second point, all 225 paths proceed to the terminus, (node H, at location 1.0, 1.0). Figure 5.5 shows just four of the possible paths.

The AUC and probability of taking each path can be calculated for a given set of ROC data.

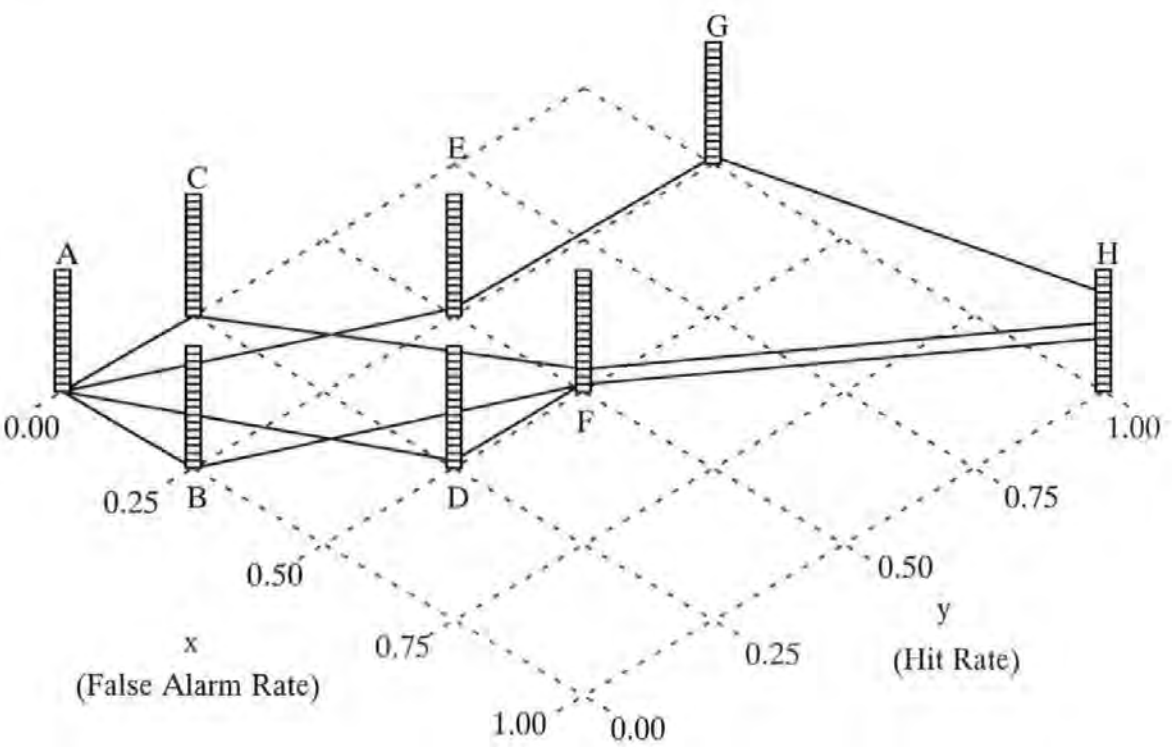


Figure 5.5 Factorising AUC paths

The propability, p, of each path (expanding equation 3.4 for two points) is:

$$p \propto {x_0}^{a_0}\, {y_0}^{b_0}\;\; {x_1}^{a_1}\, {y_1}^{b_1}\;\; (1-x_0-x_1)^{a_2}\, (1-y_0-y_1)^{b_2}$$

and the AUC (by the trapezoid rule, section 3.4.1) is:

$$AUC = \frac{x_0 y_0}{2} + \frac{x_1(2y_0 + y_1)}{2} + \frac{(1 - x_0 - x_1)(y_0 + y_1 + 1)}{2}$$

If the AUCs of each path are quantized, a histogram can be constructed summing the probabilities of all paths with the same (quantized) AUC in each histogram bin, so generating an approximation of the pdf of the AUC.

While evaluating a two point ROC curve on a 5 by 5 grid gives a modest 225 paths, a more realistic grid size is at least 128 by 128. Evaluating all paths gives a calculation proportional to the square of the volume of a hypertetrahedron, ($(G^n/n!)^2$, where G is the grid size, and n the number of ROC points), which rapidly becomes unrealistic for even a modest number of points.

However, by following the breadth first search algorithm above, and summing the AUC in stages:

A → B (0/16)

A → C (0/16)

A → D (1/16)

A → E (1/16)

A → B → F (1/16)

A → C → F (3/16)

A → D → F (1/16)

A → E → G (1/16)

A → B → F → H (7/16)

A → C → F → H (9/17)

A → D → F → H (7/16)

A → E → G → H (13/16)

it can be seen that two paths (A→B→F and A→D→F) arrive at F with the same partial AUC. The last segment from F to H will multiply both probabilities up to that point by the probability of the segment F→H, and will add to both partial AUCs the partial AUC

of the segment F→H. This duplication can be saved by combining both paths at node F by adding together the probabilities of both paths up to that point.

Algebrically, the probability of the path A→B→F→H is:

$$p_{ABFH} \propto (x_B - x_A)^{a_0} (y_B - y_A)^{b_0} \cdot (x_F - x_B)^{a_0} (y_F - y_B)^{b_0} \cdot (x_H - x_F)^{a_0} (y_H - y_F)^{b_0}$$

and the probability of the path A→D→F→H is:

$$p_{ADFH} \propto (x_D - x_A)^{a_0} (y_D - y_A)^{b_0} \cdot (x_F - x_D)^{a_0} (y_F - y_D)^{b_0} \cdot (x_H - x_F)^{a_0} (y_H - y_F)^{b_0}$$

(where both are in the same proportion), therefore, by factorising, the probability of path A→B→F→H, plus path A→D→F→H is:

$$p_{ABFH} + p_{ADFH} \propto ((x_B - x_A)^{a_0} (y_B - y_A)^{b_0} \cdot (x_F - x_B)^{a_1} (y_F - y_B)^{b_1} + (x_D - x_A)^{a_0} (y_D - y_A)^{b_0} \cdot (x_F - x_D)^{a_1} (y_F - y_D)^{b_1}) \cdot (x_H - x_F)^{a_2} (y_H - y_F)^{b_2}$$

where:

x_A is the x coordinate of point A on the ROC graph;

y_A is the y coordinate of point A on the ROC graph;

similarly for x_B, y_B etc.

Note however that path A→C→F→H also passes through node F, but with a different AUC. This path can not be combined with the paths A→B→F and A→D→F. Only paths that arrive at the same node, with the same AUC, can be be combined.

This is shown on Figure 5.5 by the columns at each node, which are divided into levels, or quantized values for the AUC up to that point. Only paths arriving at the same node, on the same level, are combined. Figure 5.5 shows the paths being elevated up the levels as their AUC builds up along their journey. For instance, the path A→E→G→H climbs through the levels 0/16 → 1/16 → 1/16 → 13/16 along its journey.

5.4.1 Software Implementation

The algorithm, implemented in C++, uses a 4D data structure with one element for each quantized partial AUC, for each quantized x and y location.

The first step is to calculate the quantized partial AUC (level) of the first ROC point at every x, y coordinate, calculate the probability of that path, and store the probability at the AUC level for that x, y coordinate.

The next step is to iterate across all x and y start values, and for each start value, to iterate across all possible x, y end values for that segment, calculating the probability of the path and the partial AUC of that segment. Each element of the column of partial AUC probabilities is multiplied by the probability, the whole column shifted up by the quantized partial AUC, and added to the column of partial AUC probabilities at the end x, y coordinate. The whole process is then repeated for the next segment of the path, until the penultimate stage.

In the final stage, the partial AUC columns are gathered up into the final AUC histogram by multiplying each element of each column by the probability of its final segment, shifting the whole column up by the partial AUC of its final segment, and accumulating them into the final histogram. Each element of the final histogram is then normalized by dividing by the sum of the elements.

The C++ code for the algorithm is given in appendix section C.3.

It should be noted that the AUC columns only need to be as tall as the maximum value of the partial AUC at that x, y coordinate (e.g. the column at location E only needs to be 2 elements tall), which gives more efficient use of memory.

The memory use of this algorithm is proportional to $G_x\ G_y\ G_a/4$, and the run time is proportional to $G_x^2\ G_y^2\ G_a\ n$. Where G_x is the grid size in the x direction, G_y is the grid size in the y direction, G_a is the grid size of the AUC pdf, and n is the number of ROC points.

As this algorithm samples, rather than integrates the pdf of the AUC, the False Alarm Rate is actually sampled at values of 0, $1/m$, ... m/m, where $m + 1 = G_x$. The same is done

for the Hit Rate. This should be contrasted with the integration of the pdf of ROC points as discussed at the end of section 5.2.7. In order to minimise rounding errors using the precalculated table of powers, (see section 5.2.7.2) m was chosen to be a power of 2, thus G_x (and G_y) is an odd number.

5.5 Parametric ROC Curves

As was explained in section 3.5, the two parameters of a binormal ROC curve, $\Delta\mu'$, σ_h, are uniquely specified by a two point ROC curve. The probability of those two points generating the sample a_0, a_1, a_2, b_0, b_1, b_2, is given by:

$$p_{xxyy} \propto x_0^{a_0} \; x_1^{a_1} \; (1 - x_0 - x_1)^{a_2} \; y_0^{b_0} \; y_1^{b_1} \; (1 - y_0 - y_1)^{b_2}$$

The pdf of this function is constrained to a 4 dimensional hyperplane in a 6 dimensional hypercube. Each point on this 4 dimensional hyperplane has a unique mapping to a point in $\Delta\mu'$, σ_h parametric space, and thus the pdf in x, y space can be mapped to a pdf in $\Delta\mu'$, σ_h space.

The same mapping principle can be applied to map the pdf in x, y space to the pdf in $\Delta\mu'$, σ_h space for any number of points. However, the $\Delta\mu'$, σ_h parameters are uniquely specified by just two points. The third and subsequent points must either fall on the parametric ROC curve, or the curve is nonparametric.

The same factorising principle can be applied as was used for the nonparametric AUC. Instead of using a column of quantized AUC values at each point, a grid of $\Delta\mu'$, σ_h values was used. Now if the path A→B→F→H (Figure 5.6) has the same parametric parameters as path A→D→F→H their probabilities can be summed by factorisation as before.

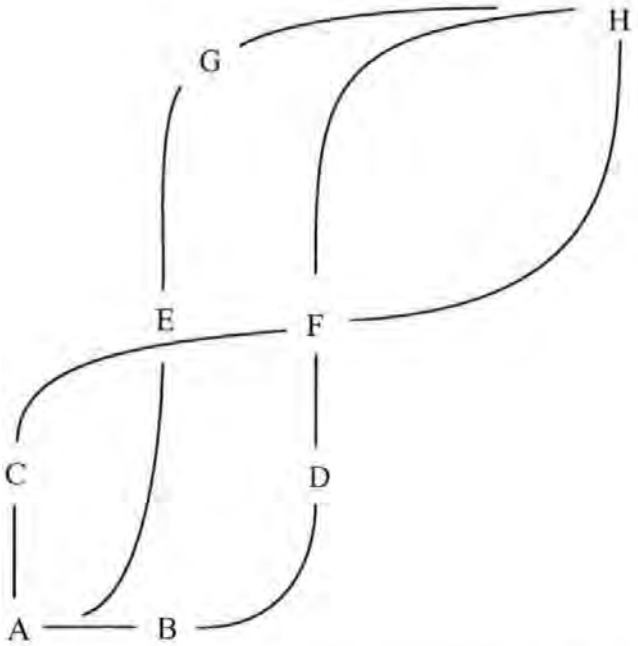


Figure 5.6 Paths of parametric ROC curves

Each path segment is a two point ROC curve with a start and end point (x_{i-1},y_{i-1} to $x_{i-1} + x_i$, $y_{i-1} + y_i$, for the i^{th} point) on the ROC graph which has a uniquely mapping to $\Delta\mu'$, σ_h space. If the second segment of a path does not have the same parameters as the first segment of the path, the path is nonparametric, similarly for the third segment etc. In other words, a valid parametric path must have the same parameters for every segment. If a path 'falls off' a parametric line it becomes nonparametric, and its probability can be summed with the nonparametric paths.

The algorithm thus generates a pdf in a quantized $\Delta\mu'$, σ_h grid, with an extra element for the nonparametric probability.

The pdf of the AUC can then be generated as a mapping from the pdf in $\Delta\mu'$, σ_h space.

5.5.1 Software Implementation

A direct implementation of the algorithm described as above would use a 4 dimensional data structure. Each x, y point on the grid would require a grid of all possible $\Delta\mu'$,σ_h values. For every segment move a $\Delta\mu'$, σ_h value would be calculated for the segment, and the probability in the start $\Delta\mu'$, σ_h location would be multiplied by the probability of the move ($x_i^{a_i} \cdot y_i^{b_i}$), and added to the destination $\Delta\mu'$, σ_h location.

However, a realistic grid size is about 256 elements a side. A 4 dimensional data structure of 256 elements in each dimension is 4 giga–elements (256^4) in size. Since a probability is stored as either an 8 byte double variable, or a 12 byte REAL variable (section 5.2.7.1), this requires 32 gigabytes, or 48 gigabytes of memory, respectively. The memory of a PC is limited by its address space to 4 gigabytes.

In addition, every move requires $\Delta\mu'$, σ_h to be calculated, which involves a look–up in the inverse standard normal distribution, a few floating point multiplies, and then a function evaluation to compute the sigmoid mapping of $\Delta\mu'$.

The solution is to note that each path must have the same $\Delta\mu'$, σ_h value for every segment. Any diverging path is nonparametric. Therefore the paths for each $\Delta\mu'$, σ_h value are totally independent of any other $\Delta\mu'$, σ_h paths and can be calculated separately. This means the data storage only has to contain the x, y grid with the current $\Delta\mu'$, σ_h data, and the algorithm can then iterate over all values of $\Delta\mu'$ and σ_h.

For any $\Delta\mu'$, σ_h pair, only a fraction of the possible segments, from any given x, y point to any other x, y point, are valid. It would be a massive computation to iterate over all segments calculating the valid ones, so a look–up table of valid segments for every $\Delta\mu'$, σ_h pair is stored. A grid of pointers, indexed by $\Delta\mu'$, σ_h, is used to point to the corresponding blocks of valid segments. (The length of each block is also stored in another grid indexed by $\Delta\mu'$, σ_h). Of course, this reintroduces the storage problem. Every possible segment is equivalent to a two point ROC curve, and uniquely maps to a $\Delta\mu'$, σ_h pair, so every possible segment is in the valid block of precisely one $\Delta\mu'$, σ_h pair. The only salvation is that now only x, y coordinates need to be stored. One byte per coordinate was chosen, which because the $\Delta\mu'$, σ_h grid is symmetrically arranged around zero, gave a range of $\pm$ 64 for $\Delta\mu'$ and σ_h. This in turn reduced the segment storage to 129^4 segments which, with 4 coordinates per segment (the (x, y) coordinates of the beginning and end of each segment), requires 1.03 Gigabyte of memory. A mesh size of 129 is far from ideal, but was a pragmatic choice within the limitation of the computer resources available.

The set of all possible segments include the zero length segments where a ROC point occurs at the same location as the previous one (x_{i-1},y_{i-1} to $x_{i-1} + x_i$, $y_{i-1} + y_i$, where $x_i = 0$, $y_i = 0$). Two ROC points with zero separation give an undefined binormal ROC curve. Defining such segments as nonparametric was considered, but this is not really valid either. It was therefore decided to assign these zero length segments to the nearest parametric ROC curve. This has the advantage of keeping the code for calculating the pdf fast and simple. Since finite size grids are used, there is not a precise mapping from x, y space to $\Delta\mu'$, σ_h, space – there are always rounding errors – and so a 'closest' ROC curve can always be defined for any x, y point on the ROC graph. These points are only important when the corresponding data values (a_i b_i) are both 0, in which case the probability of the segment, raised to the power of the data value, ($x_i^{a_i} \cdot y_i^{b_i} = 0^0 \cdot 0^0 = 1$), propagates the probabilities down the path to the next segment. Any other value of either a_i or b_i zeroises the propagated probability. The other point to consider, is that the larger the grid size, the smaller the influence of even this small effect.

Under this scheme the nonparametric probability is ignored, and for most purposes it is not required anyway, but the Monte Carlo simulation to validate the method does require it (section 6.6). The probability is thus calculated nonparametrically by iterating over the grid using all segments. (The block of valid segments for the nonparametric paths includes all segments). However, this nonparametric value includes the parametric values as well, so they are subtracted. The result is a $\Delta\mu'$, σ_h, pdf grid with an extra element for the nonparametric component.

The memory use is proportional to G_x^2 G_y^2, and the run time proportional to G_x^2 G_y^2 n, where G_x is the grid size in the x direction, G_y is the grid size in the y direction, and n is the number of ROC points. Again, this algorithm samples the pdf, and so G_x and G_y are odd numbers (see last paragraph of section 5.4.1).

The program produces a plot of the 95% posterior interval of the $\Delta\mu'$, σ_h parameters, a plot of the AUC giving the 95% posterior interval, and a text file with the 95% posterior interval of the AUC.

5.6 Confusion Matrices

5.6.1 Unweighted Confusion Matrix

		Standard		
		A	B	None
Test	*A*	a_0	a_3	a_7
	B	a_5	a_1	a_4
	None	a_8	a_6	a_2

Figure 5.7 Unweighted confusion matrix

Figure 5.7 (a copy of Figure 1.7) shows a three by three unweighted confusion matrix where the three cells along the diagonal (shaded in grey) are correct classifications, and the other six cells (shown in white) are incorrect classifications. The number of samples in each cell are given by a_0, a_1, ..., a_8. The population probability of each cell can then be given by x_0, x_1, ..., x_8 respectively.

Suppose an intelligent medical system is tested with a sample of 20 cases and 18 results are on the diagonal and 2 are in the rest of the matrix. It does not actually matter where the 2 misclassified cases are. It may be that a_3 is 1 and a_6 is 1, or a_7 could be 2. All these misclassifications are of equal consequence, and the system performance is the same. (If these misclassifications are not of equal consequence then a weighted confusion matrix should be used). An unweighted confusion matrix can thus be seen as a binary classifier which gives system performance in terms of the fraction of correct results, or rather, as the pdf of the fraction of correct results. Posterior intervals can be produced from the pdf.

In order to calculate the pdf, the individual population probabilities of each cell are not required, all that is needed is the probability of correct classifications i.e. the sum of probabilities $s = x_0 + x_1 + x_2$ irrespective of the individual probabilities x_0, x_1 or x_2. (The misclassification probability is hence given by $1 - s = x_3 + x_4 + x_5 + x_6 + x_7 + x_8$). This problem has already been analysed and solved in section 5.2 and the general form of the solution is given by equation 5.16:

$$f(s) = s^{\sum_{i=0}^{n-1} a_i + n - 1} \; (1 - s)^{\sum_{i=n}^{m} a_i + m - n}$$

Here, n is now the number of the cell on the diagonal, and m the total number of cells in the matrix. Thus, the order term is derived from the number of cells in the matrix (assuming a uniform Bayesian prior). Expanding equation 5.16 using the example in Figure 5.7 gives:

$$f(s) = s^{a_0 + a_1 + a_2 + 2} \; (1 - s)^{a_3 + a_4 + a_5 + a_6 + a_7 + a_8 + 5}$$

Equation 5.16 is a general form of the False Alarm Rate term of equation 5.2 (section 5.2.5) from which the linear pdf can be approximated by a vector (and then plotted) by using equation 5.13 to calculate the vector elements.

This analysis is actually identical to the analysis of the False Alarm Rate of a ROC curve and it is for this reason that the same variables a_0, a_1, ..., a_8, and x_0, x_1, ..., x_8 have been used.

5.6.2 Weighted Confusion Matrix

		Standard		
		A	B	None
Test	*A*	a_0	a_3	a_7
	B	a_5	a_1	a_4
	None	a_8	a_6	a_2

Figure 5.8 Weighted confusion matrix

Figure 5.8 (a copy of Figure 1.8) shows a weighted confusion matrix with three weights. Correct classifications along the diagonal are shown in grey. Misclassifications of minor consequence are shown in light grey, while misclassifications of major consequence are shown in white. Suppose a system is tested with a sample of 20 cases and 18 results are on the diagonal, 1 is a minor misclassification and 1 is a major misclassification. It does not matter if the minor misclassification is cell a_3 or cell a_4, or whether the major miscala-saification is cell a_5, a_6, a_7 or a_8, because the overall performance of the system is the

same. A weighted confusion matrix can thus be seen as an n–ary classified according to the number of different types of weighted classifications it has. The example in Figure 5.8 has three weighted classifications.

In a binary classifier, i.e. an unweighted confusion matrix, the pdf is plotted by integrating and vectorising equation 5.16, as discussed above. In equation 5.16 there are only two terms, a term for the number of correct classifications, and a term for the number of misclassifications. The correct classification term is raised to the power of the sum of the number of correctly classified samples, plus the order term (which is 1 – the number of correct classification cells). The misclassification term is raised to the power of the sum of the number of misclassified samples, plus the order term (which is 1 – the number of misclassification cells).

A hypothesis will now be introduced that this form of equation can be logically extended to any number of terms, and hence any number of weighted classifications. Applying this extension to the example in Figure 5.8 gives:

$$f(s') = {s_0}^{a_0+a_1+a_2+2}\ {s_1}^{a_3+a_4+1}\ (1 - s_0 - s_1)^{a_5+a_6+a_7+a_8+3} \tag{5.18}$$

where s_0 is the population proportion of correct classifications, s_1 is the population proportion of minor misclassifications, and $s_2 = 1 - s_0 - s_1$ is the population proportion of major misclassifications. The order term is the number of cells in each term, minus one. This describes a pdf in two dimensions, which is actually a plane across the diagonal of a cube with axes s_0, s_1, s_2 as shown in Figure 5.9.

Each point on the surface represents a different proportion of correct classifications, minor misclassifications, and major misclassifications, each of which has a weight. Thus the weight at any point on the surface can be calculated by the sum of the contributions to the weight by each classification. The weight, W, at any point on the surface of Figure 5.9 is thus given by:

$$W = s_0 w_0 + s_1 w_1 + (1 - s_0 - s_1) w_2$$

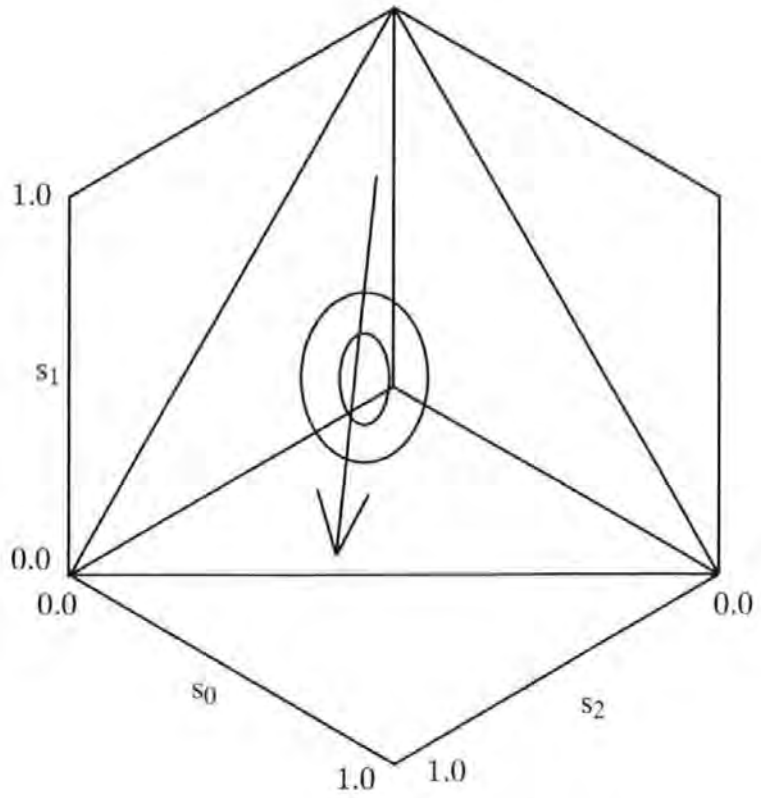


Figure 5.9 Pdf of 3 weight confusion matrix showing isoweight arrow

Where w_0 is the weight of correct classifications, w_1 is the weight of minor misclassifications and w_2 is the weight of major misclassifications. The probability that a point on the surface produces the sample is given by $f(s')$ (equation 5.18).

If the probability of every point on the surface that has the same weight could be summed the pdf of the weight could be produced. If $w_0 = 1.0$, $w_1 \approx 0.3$, and $w_2 = 0.0$, then the arrow on Figure 5.9 represents a line of equal weight (an isoweight contour). Integrating over the plane parallel to the arrow would therefore give the pdf of the weight.

The interpretation of the pdf of the weight depends on the meaning of the weights themselves. Section 2.2 discussed possible measures of clinical performance, e.g. QALYs, micromorts etc., that could be used as weights. The weights could also be fiscal costs. Assuming the weight is a saving in pounds sterling (£), the pdf of the weights gives the pdf of the saving, from which the posterior interval of the saving can be derived.

Continuing with the example above, 18 cases were correctly classified, which saves £ 18, there was one minor misclassification, saving £ 0.30, and one major misclassification sav-

ing £ 0.00. Thus the average saving of the sample is £ 0.915 per patient. By calculating the pdf of the weight (saving) the posterior interval of the saving in the population can also be estimated.

The pdf can be calculated by another variation of the factoring algorithm. The algorithm requires a rectangular grid of quantized weight along one axis, and by quantized probabilities along the other. Figure 5.10 illustrates this grid.

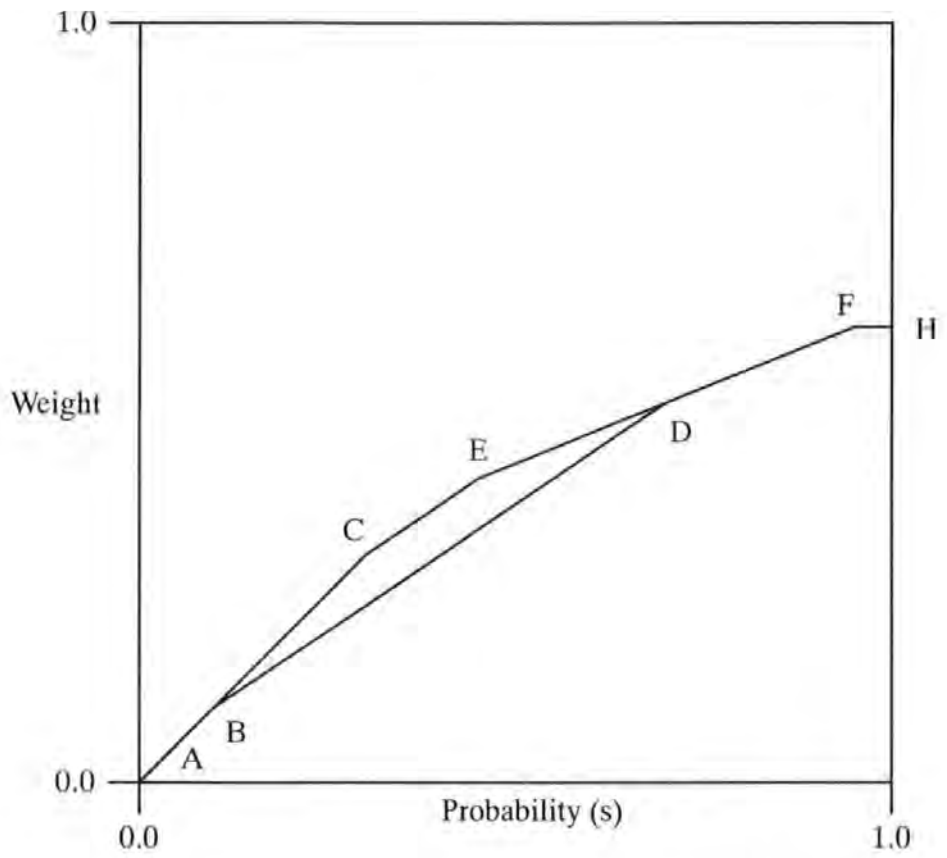


Figure 5.10 Factorising weighted confusion matrix paths

It is convenient to translate and scale the weights so they are normalized to the range 0 to 1, and order the weights w_0 to w_n in descending order. A confusion matrix with four weights is required to demonstrate factoring. The weights $w_0 = 1.0$, $w_1 = 2/3$, $w_2 = 2/5$ and $w_3 = 0.0$ are convenient.

Assuming the grid is quantized into 21 by 21 elements, s_0 can take any of the values 0.00, 0.05, ..., 1.00. For each possible value of s_0, s_1 can be any value from 0 to $1 - s_0$. For

instance, if s_0 is 0.65, s_1 can take any of the values 0.00, 0.05, ..., 0.35. For each possible combination of s_0 and s_1, s_2 can be any value from 0 to $1 - s_0 - s_1$. Finally s_3, must be equal to $1 - s_0 - s_1 - s_2$. There are thus $21^4/4!$ possible paths (the volume of a hypertetrahedron). For each path, both the weight, and the probability of generating the sample, can be calculated. Factoring is used to reduce the combinational explosion of paths. Consider two paths A→B→D→F→H and A→C→E→F→H shown on Figure 5.10 where the values of s_0, s_1, s_2, s_3 are 0.1, 0.6, 0.25, 0.05 and 0.3, 0.15, 0.5, 0.05 respectively.

The first path starts at node A. Its first segment is of length 0.1 and the first weight is 1.0. The contribution to the total weight of the path is thus $0.1 \times 1 = 0.1$. The segment thus ends at node B at location (0.1, 0.1) (given in the format: proportion, weight). The next weight is 2/3, and the path has a length of 0.6, which adds $0.6 \times 2/3 = 0.4$ to the weight contribution, giving node D at (0.7, 0.5). The third weight is 2/5, and the segment is 0.5 long. The segment therefore adds $0.25 \times 2/5 = 0.1$ to the weight contribution and ends at node F, at location (0.95, 0.6).

The second path also starts at node A. It has a first segment of length 0.3 and a weight of 1. The segment therefore ends at node C at location (0.3, 0.3). The second segment has a length of 0.15 and weight of 2/3, which therefore adds $0.15 \times 2/3 = 0.1$ to the combined weight and ends at node E at (0.45, 0.4). The third segment has a length of 0.5, and therefore $0.5 \times 2/5 = 0.2$ is added to the weight. The third segment ends at node F. at location (0.95, 0.6).

Both of these paths therefore meet at node F where the probability that each of these paths generated the sample can be combined. The last segment has a weight of 0, and a length of 0.05 (the total length of every path must be 1.0). The fourth segment thus contributes $0.0 \times 0.05 = 0.0$ to the weight and terminates at node H at location (1.0, 0.6).

This algorithm has used the hypothesis that cells with the same weight can be combined provided account is taken of the order terms. This follows from the analysis in section 5.2 that derived an expression for the one dimensional pdf obtained by integrating the pdf

lying on an $n-1$ dimensional hyperplane in an n dimensional hypercube at right angles to the axes. The confusion matrix algorithm assumes a pdf on an $n-1$ dimensional hyperplane can be integrated to produce a pdf on a $w-1$ dimensional hyperplane, (where w is the number of unique weights) which can then be summed across different slices (not at right angles to the axes) by the factoring algorithm.

(a)

Cases	Weight
4	0.0
3	0.3
1	0.3
6	1.0

(b)

Cases	Weight
4	0.0
5	0.3
6	1.0

Table 5.1 Confusion matrix data that gives the same pdf

This has been investigated with informal experiments. For example, the weighted confusion matrix data in Table 5.1 (b) can be derived from that in Table 5.1 (a) be combining the number of cases with a weight of 0.3 and adding one for the order term. Figure 5.11 shows the pdf of the weighted confusion matrix in (a) plotted on the same graph as the pdf of the weighted confusion matrix (b). It can be seen that the two pdfs are identical, and hence they are superimposed.

It should be noted that these two sections on unweighted and weighted confusion matrices have not mentioned chance correction which is a defining feature of the kappa statistic. Change correction for the pdf will be discussed in section 8.3.5.

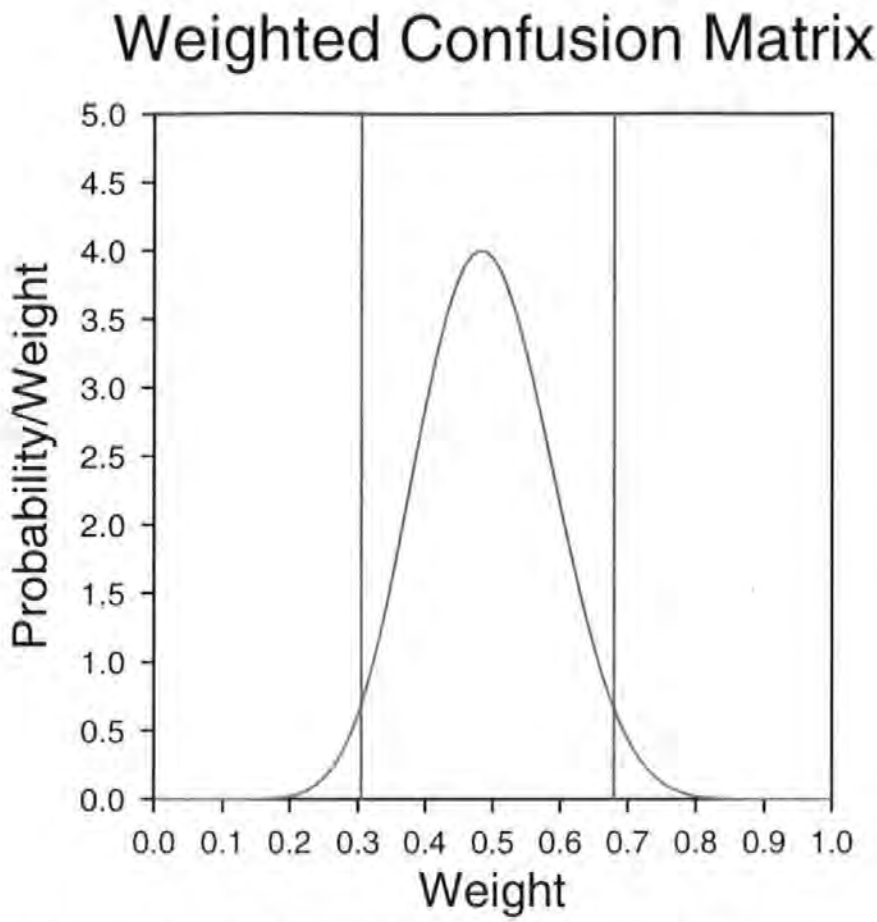


Figure 5.11 Data in Tables 5.1 (a) and (b) give the same pdf

5.6.3 Software Implementation

A program was written in C++ to perform the calculations and plot the pdf of the weight and the 95% posterior interval. The values of the posterior interval was also written as text to a file.

The program input requires that the numbers of cases in cells with the same weight are aggregated together and the order term is added. The input is thus a list of weights and the total number of cases (plus order term) with that weight. This format allowed the informal experimentation discussed above to be undertaken, where Table 5.1 (a) and (b) give an identical pdf, as shown in Figure 5.11.

The memory use of the algorithm is proportional to $G_w \; G_x$, and the run time proportional to $G_w \; G_x \; n$, where G_w is the grid size in the W direction, G_x is the grid size in the x direction, and n is the number of weights.

5.7 Summary

This chapter has presented novel methods for calculating the posterior intervals of ROC curve and confusion matrix statistics.

An analytic method was used to calculate the pdf, and from it, the posterior interval of each point of a nonparametric ROC curve. The analysis reveals that an order term emerges as a consequence of using a uniform Bayesian prior distribution for multi-point curves. This may presents a problem with deciding the number of categories to use for a ROC analysis a priori if an uninformative prior cannot be used because of categories with zero cases.

A method of comparing the pdfs of uncorrelated nonparametric ROC points was explained. Such comparisons can be used in the objective comparison of system and expert performance.

An algorithm for calculating the posterior interval of the Area Under the Curve (AUC) of a nonparametric ROC curve was then presented. The algorithm works by factorising all the possible (quantized) ROC curves that could possible generate the sample. An adaptation of the algorithm was used to generate the posterior interval of the AUC of a parametric curve, and to present the posterior interval of the parameters.

Finally, another adaptation of the factoring algorithm was used to derive a algorithm for generating the posterior interval of the weight of a weighted confusion matrix. Provided objective weights are used in the confusion matrix, the posterior interval of the weights will have an objective meaning.

All of these methods have been implemented in software. Chapter 6 describes how this software was tested with Monte Carlo simulations to determine if the posterior intervals are robust and accurate, while Chapter 7 demonstrates the use of the methods on examples from the literature, illustrated by graphs produced by the software.

6 Monte Carlo Simulations

This chapter discusses the Monte Carlo simulations used to test all the algorithms documented in Chapter 5.

6.1 General Simulation Plan

All the Monte Carlo simulations followed the same basic plan. A population point was randomly generated. Then a sample of cases was generated from the population point. This synthesised data was used to generate the probability density function of the sample. The position of the population point within the probability density function generated from the sample was then recorded, and used to produce a histogram of 20 bins, giving the number of times the population points fell in the 5%, 10%, ... 95%, 100% posterior interval of the sample. Each test was run a large number of times, with the expectation that the same number of points would be found in each of the 20 posterior intervals. A chi-squared (χ^2) measure (section 4.2) was taken of the results, and the experiment run repeatedly to obtain a histogram of the chi-squared values. If an experiment was working, each histogram of chi-squares would approximate the chi-squared distribution for 19 degrees of freedom.

Plots of the chi-squared histograms of selected results are shown in this chapter. The theoretical chi-squared distribution was plotted as a curve on each histogram so that a visual comparison can be made between the results expected in theory, and those obtained in practice. The full results of each experiment are shown in more compact tables in the Appendix B. A working experiment should result in half the chi-squared tests having a value of 18.34 or less.

The different algorithms require widely differing computer resources, so while some Monte Carlo simulations were run 818,400,000 times in a matter of days, other have only been run 36,000 times, in simulations that took weeks. These limited results are regarded as pilot studies.

6.1.1 Random Number Generator

In order to generate a pseudo–random number sequence as close to a true random number sequence as possible the L'Ecuyer [88] pseudo random number generator, with a period in excess of 2×10^{18}, incorporating a Bays–Durham shuffle with added safeguards was used for all simulations.

6.1.2 Linear Interpolation on Contour Boundaries

During the Monte Carlo experiments there was always the possibility that a population point would be generated at the boundary between two posterior intervals.

For the purposes of illustration, assume there are only two posterior intervals, (i.e. the 50% and 100% posterior intervals) and that the pdf is divided into eight elements or 'tiles' (labelled *a* to *h*). One of the 'tiles' is chosen at random as the population point. A sample is generated from the population point, and the pdf of the population point calculated from the sample. (I.e. the probability that each 'tile' contains the population point). The 'tiles' are sorted in order of probability, as shown in Figure 6.1. In this example, it can be seen that 'tiles' *a* to *e* are below the 50% posterior interval (PI), and tiles *g* and *h* are above. If the population 'tile' that generated the pdf was *c*, then the 'tile' is in the 50% PI. If the 'tile' was *g*, then the 'tile' is in the 100% PI. However, if the population point is 'tile' *f* it is not clear which PI the 'tile' belongs to. Linear interpolation was therefore used to calculate the fraction of each 'tile' in each PI and the PI chosen at random using this fraction as a probability. In Figure 6.1, 'tile' *f* is $3/19^{th}$ in the 50% PI and $16/19^{th}$ in 100% PI.

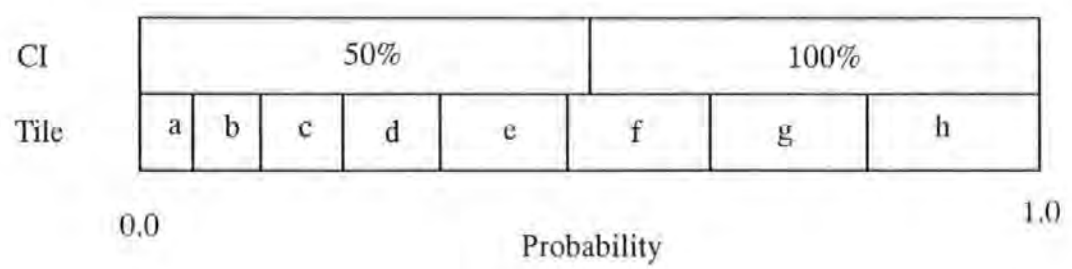


Figure 6.1 Interpolation of posterior intervals

In Figure 6.1, the change in probability across 'tiles' is a constant, and therefore linear interpolation is ideal. It is less accurate when the change in probability is increasing or

decreasing (i.e. at points of high curvature of the pdf). This happens when the sample size is large. The higher the sample size the more the pdf tends towards a single spike, and therefore the less resolution a fixed grid size has around the spike. The granularity of the grid therefore limits the maximum sample size at which a Monte Carlo experiment will work.

6.1.3 Simulating the Whole Population Space

It should be noted that these Bayesian simulation studies are unusual in the method of picking the population point. This can be illustrated by comparing the method with that used for ROC curve simulation studies in the literature. Frequentist studies (see sections 3.6.2) use the following procedure:

- Fix the parameters of the curve, e.g. to a binormal curve with an AUC of 0.8;
- Generate random data samples from that population curve;
- Generate sample ROC curves from the data;
- Verify that the posterior/confidence limits (e.g. 95%) of the sample curves, contains the population curve, the correct percentage of the time.

The current Bayesian study used the following procedure:

- Simulate population ROC points occurring anywhere on the surface;
- Generate data samples from the point;
- Plot the probability density function from the data sample;
- Verify that the point was within given percentiles of the probability density function the correct percentage of the time.

The method used for this study will not work when the parameters of the curve are fixed. This can be explained by considering the following Gedankenexperiment. Figure 6.2 shows the probability density function of a ROC point as a contour map, with the 33%,

66%, and 100% contour marked. The 100% contour covers the whole graph. The interpretation of the contours, is that 33% of the population points that might have produced this sample, are inside the 33% contour, 33% of the points are between the 33% and the 66% contour, and 34% are between the 66% and 100% contour. For the sake of the Gedankenexperiment, the contours should be regarded as steps with uniform density within each contour. Now consider running a Monte Carlo experiment, where the sample that produced this probability density function happens to be generated 100 times. If the contours are correct, about 33, 33 and 34 population points will originate from within each contour respectively. Now consider only generating the test population ROC points from the grey area which shows a region of hypothetical ROC curves. The easiest way to do this, is to regard the grey area as a mask. If the experiment is repeated with the mask, only generated population points that happen to lie in the grey area are used. Since the whole of the 33% contour is grey, about 33 population points will originate from the 33% contour. However, the 66% contour is only about 40% grey, so 60% of the cases generated in this area are masked out, leaving about 13 population points in the contour. Similarly, the 100% contour is only about 10% grey, so 90% of the population points will get masked out, leaving about 3 population points. A chi-squared test against the expected result of 33, 33, 34 will therefore fail. In other words, there is no point in a posterior interval including any area that cannot have produced a population point, or conversely, all points on the surface have to be able to produce a population point. The Gedankenexperiment can be extended to the situation where there are multiple 'grey' regions with various probabilities of masking population points, and any number of step contours. At the limit, the mask becomes an arbitrary probability density function, and the step contour approximation becomes a smooth probability density function. It can be seen that the experiment is unlikely to work, if the distribution of population points is uneven.

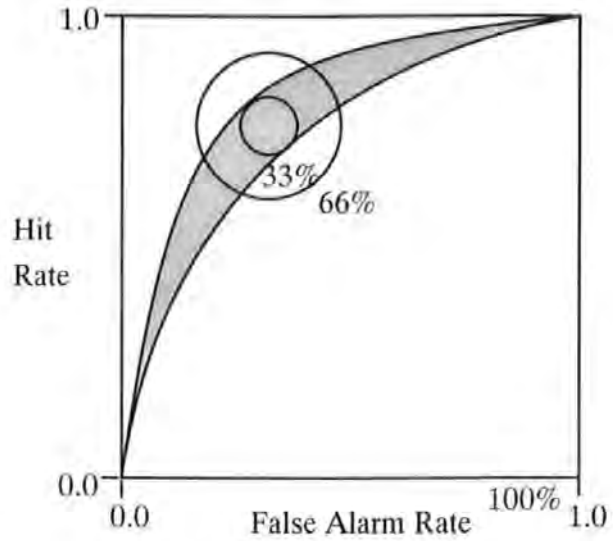


Figure 6.2 Gedankenexperiment on the distribution of ROC curves

6.2 Nonparametric ROC Points

For the nonparametric ROC point experiment the grid size was set to 512 × 512. Samples with 2^n, n=0..10 cases were simulated. Each sample size was simulated with a different set of frequencies of disease in the population. Samples with one case, were simulated with a frequency of disease of 1/2, samples with two cases with frequencies of disease 1/2 and 1/4, through to samples with 1024 cases being simulated with frequencies of $1/2^n$, n=1...11. It was not considered worthwhile simulating situations where the frequency of disease, in relation to the number of cases, would often result in no diseased cases at all. For each sample size, at each frequency of disease, ROC curves with 1, 2, 4, 8 and 16 points were simulated. Each point of the multi point ROC curves were simulated independently of the other points. This was to avoid correlation effects, as each 2D probability density function is a different view of the same multidimensional probability density function of all the ROC points combined.

For each test, a population Hit Rate and False Alarm Rate were generated for all points, whatever the actual point under test. Then each event within the sample was randomly assigned to the diseased or healthy groups, according to the frequency of disease in the population, and categorised according to the previously generated population Hit Rates and False Alarm Rates. For instance, a 4 point ROC curve has 5 categories. Each test was

run 2,000 times, with the expectation that about 100 points would be found in each of the 20 posterior intervals. The experiment was repeated 200 times to obtain a histogram of the chi–squared values. The total simulation thus generated (1 + 2 + ... + 11) × (1 + 2 + 4 ... + 16) chi–squared histograms of 200 × 2,000 simulated ROC curves, a total of 818,400,000 simulated curves, and ran for 38 hours on a 1.4 GHz Athlon PC.

A considerable saving in computer time was achieved by optimising the calculation of pdfs. The pdf of the Hit Rate is uniquely determined by the number of true positives and false negatives. Similarly the False Alarm Rate is uniquely determined the number of false positives and true negatives. This is assuming the order terms have been added as appropriate. With a maximum sample size of 1024, and a maximun order term of 15, there were only 1039 × 1038 / 2 = 538,241 possible pdf required by the experiment. Because the sample size increased by powers of two, many pdfs were never used. In order to speed the simulation, pdfs were calculated on demand and then cached for reuse if the same number of true positives and false negatives (or false positives and true negatives) occured again. This optimisation saved considerable computer time, but did require about 1 Gbyte of memory.

The population Hit Rate and False Alarm Rate for multi point ROC curves were produced by generating a uniformly distributed random number between 0 and 1 for the population Hit Rate of each point and sorting them into ascending order (i.e. simulating the uniform Bayesian prior). The same was done for the False Alarm Rate. The Hit Rates and False Alarm Rates were then paired together in the sorted order.

This is equivalent to a uniform distribution of ROC points in a hypercube of all possible points. Consider Figure 6.3 where all possible values of the Hit Rate of a two point ROC curve can be plotted (h_0 and h_1). The value of h_1 must be greater or equal to the value of h_0 for a ROC curve to be valid. Therefore only the grey upper triangle is valid. However, a point in the lower white triangle will become valid if the axes are swapped (as shown in Figure 6.3), which is equivalent to sorting the pair of randomly generated Hit Rate values. In higher dimensions the same process applies. If there are n points, there are $n!$

possible sequences of unordered Hit Rates, while the valid hyper-tetrahedron of ordered Hit Rates has a volume of $1/n!$.

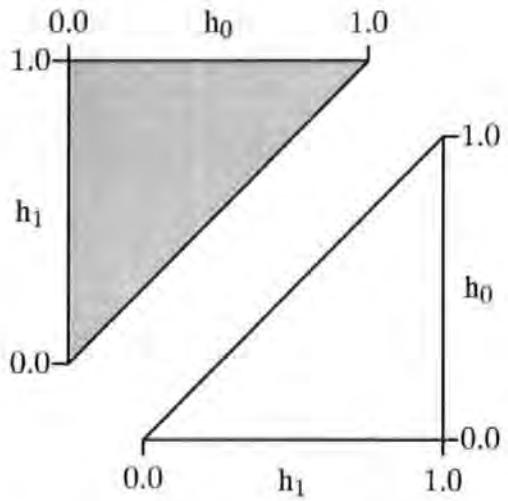


Figure 6.3 Generating a uniform distribution

This produced data compatible with the ROC curve format, but without parametric assumptions.

During test runs of the experiment it was observed that the grid size should be at least half the sample size to get good results, probably for the reasons discussed in section 6.1.2. Because sample sizes up to 1024 were tested, a grid size of 512 × 512 was used.

6.2.1 Nonparametric ROC Point Results

The simulation produced 2046 histograms in total, which gave 61 pages of tables. For brevity, only those for a one point ROC curve, the 5th point of an 8 point ROC curve, and the 1st, 8th, and 16th points of a 16 point ROC curve are shown in appendix Table B.2. Figure 6.4 shows 21 histograms for the 5th point of 8 point ROC curves over 6 sample sizes, and 6 frequencies of disease. The theoretical chi-squared distribution is plotted as a curve on each histogram so that a visual comparison can be made between the results expected in theory, and those obtained in practice. As illustrated by the diagrams, the experimental results show the expected chi-squared distributions. Given the number of cases simulated, 400,000 in each histogram, this indicates that the method is working within the limitations of the quantization of the ROC graph, into 512 × 512 elements and the stochastic nature of Monte Carlo simulations.

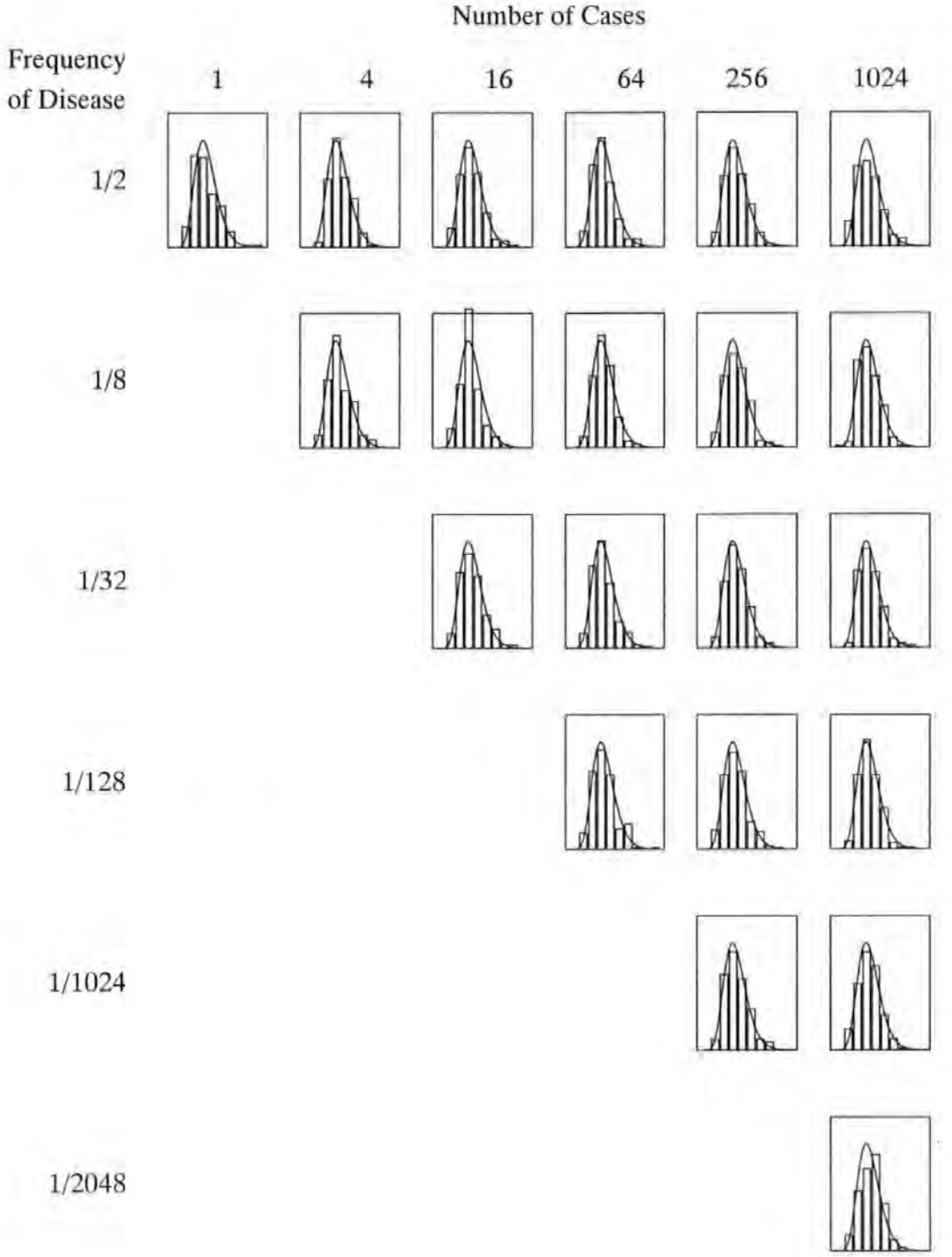


Figure 6.4 Histograms for 5th point of an 8 point ROC curve

6.3 Fixing the Population Point

A fixed population point was also tested to compare and contrast the two methods. Section 3.6.2 has already discussed the theoretical limitations of fixing the population point (a Frequentist method) in a Bayesian Monte Carlo simulation with a small sample size, which

predicts the experiment would fail. Section 6.1.3 above argues for using a uniformly distributed random population point, and the results of the nonparametric ROC point simulations demonstrates that this works.

One nonparametric ROC point was simulated with a Hit Rate of 0.8 and a False Alarm Rate of 0.2 using a grid size of 512 × 512. Samples with 2^n, n=0..10 cases were used. Each sample size was simulated with a different set of frequencies of disease in the population. Samples with one case, were simulated with a frequency of disease of 1/2, samples with two cases with frequencies of disease 1/2 and 1/4, through to samples with 1024 cases being simulated with frequencies of $1/2^n$, n=1...11. Each test was run 2,000 times, with the expectation that about 100 points would be found in each of the 20 contours. The experiment was repeated 200 times to obtain a histogram of the chi-squared values. The total simulation thus generated (1 + 2 + ... + 11) chi-squared histograms of 200 × 2,000 simulated ROC curves, and ran for 1.8 hours on a 1.4 GHz Athlon PC. This duplicates the nonparametric tests in section 6.2 for one ROC point.

6.3.1 Fixed Population Point Results

The results are given in appendix Table B.3. It was found that the experiment only worked when the frequency of disease was close to 1/2 and the sample size was close to 1024. If the frequency of disease dropped below 1/16, or the sample size below 512, divergence from the expected chi-squared distribution can clearly be seen. The results of six tests, for sample sizes from 32 to 1024, and with a frequency of disease of 1/2, are shown in Figure 6.5. While it was expected that these tests would fail at low sample sizes, it was surprising how high the sample size must actually be to obtain success. As a by-product, this simulation demonstrates the ability of the Monte Carlo experimental design to detect incorrect pdfs.

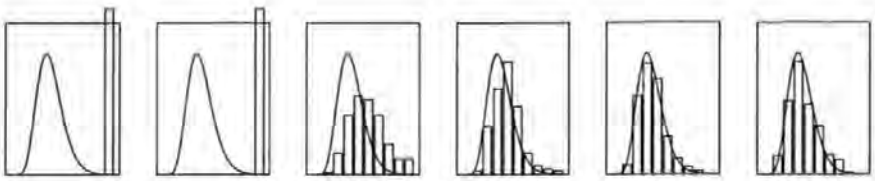


Figure 6.5 Histograms for fixed population point

It should be pointed out that a frequency of disease of 1/2 generates approximately 256 'Gold Standard' diseased cases and 256 'Gold Standard' healthy cases from a sample size of 512. It could therefore be stated that this fixed population experiment works with more than 256 diseased (or health) cases. It should also be noted that this experiment shows convergence of the Frequentist fixed population paradigm with the Bayesian paradigm at large sample sizes. This will also be shown in sections 7.4 and 7.5.

6.4 Comparing Nonparametric ROC Points

Four sets of simulations were run using different frequencies of disease for the two systems to be compared, using a ROC grid size of 512 × 512, and hence a comparison grid size of 1023 × 1023. Firstly a frequency of disease of 1/2 was run against a frequency of disease of 1/2, then 1/8 against 1/8, then 1/2 against 1/8, and finally 1/8 against 7/8. Figure 6.6 shows nine possible combinations, but it can be seen that because of duplication and symmetry (by 90 degree rotation) only the four shaded combinations are actually necessary to cover all possibilities.

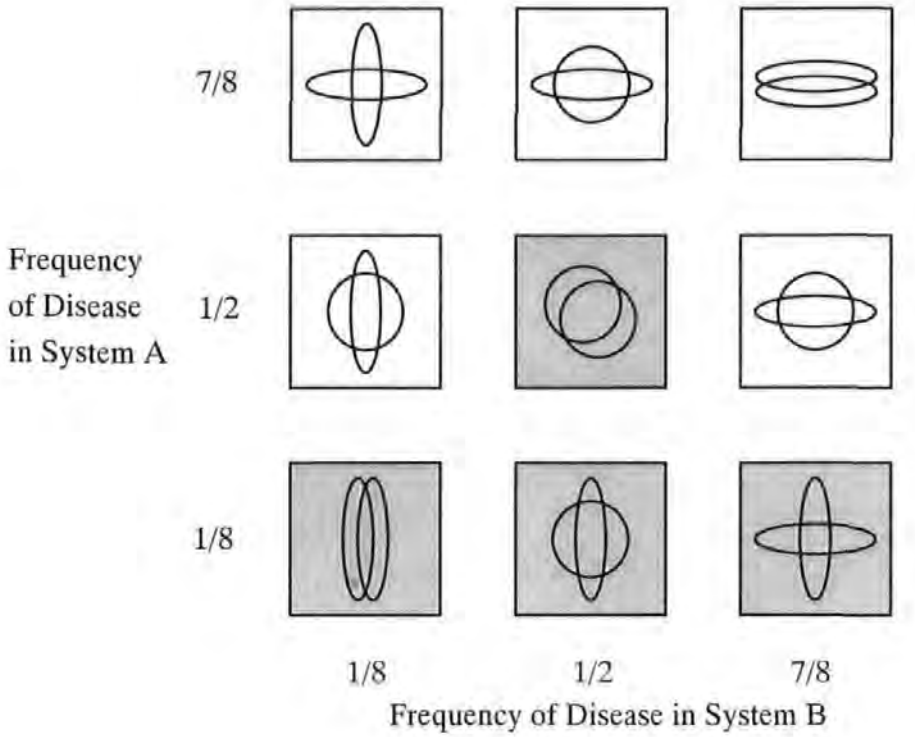


Figure 6.6 Combinations of frequency of disease

Each of these combinations was run for different sample sizes for each system. Samples with 2^{2n}, n=0..5 cases were used. The combination of different frequencies of disease in the population would obviously not occur when one system or human was being compared with another, since the same samples would be used, or at least samples from the same population, but this unlikely situation was simulated to test the robustness of the method. Similarly, a comparative test would not expect widely differing numbers of test cases, but again this was simulated to test generality and robustness. For each test, a population Hit Rate and False Alarm rate were generated for both systems. The Hit Rate pdf (Y in equation 5.14) and False Alarm rate pdf (X in equation 5.13) were generated for both systems and the pdf of the difference between systems was then calculated (equation 5.17). The difference in population False Alarm Rate and Hit Rate was calculated from the generated False Alarm Rates and Hit Rates of the two systems, and its position within the pdf of the difference recorded. Each test was run 2,000 times, with the expectation that about 100 points would be found in each of the 20 posterior intervals. The experiment was repeated 200 times to obtain a histogram of the chi-squared values. The total simulation

thus generated 4 × 6 × 6 chi-squared histograms of 200 × 2,000 simulated ROC point comparisons, and ran 24.4 hours on a 1.4 GHz Athlon PC.

6.4.1 Comparison of Nonparametric ROC Points Results

The simulation produced 144 histograms in total which are given in appendix Table B.4, of which four are shown here in Figure 6.7. One histogram is shown for each of the four different combinations of frequency of disease, at four different combinations of sample size. As illustrated by the diagrams, the experimental results show the expected chi-squared distributions. Given the number of cases simulated, 400,000 in each histogram, this indicates that the method is working within the limitations of the quantization of the ROC point comparison, into 1023 × 1023 elements, and the stochastic nature of Monte Carlo simulations.

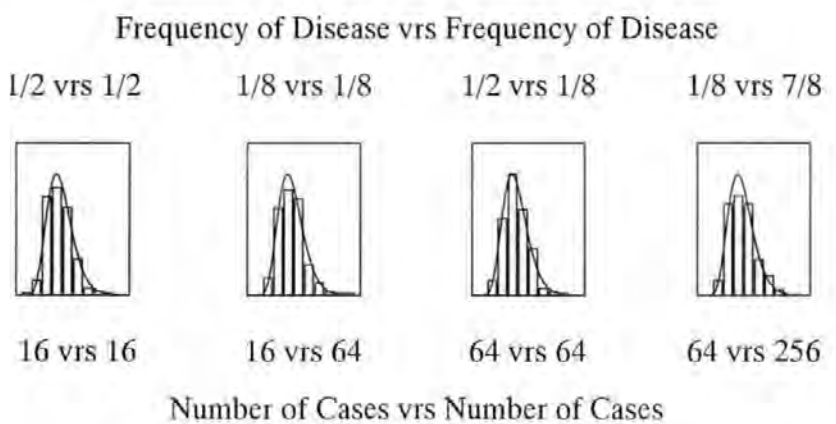


Figure 6.7 Histograms for nonparametric point comparison

6.5 Nonparametric AUC

Due to the huge computational resources required by this algorithm it was only possible to perform a preliminary experiment to demonstrate potential.

A grid size of 129 was used for the pdf of the AUC. Samples with 2^n, n=3..7 cases were simulated. The sample size was limited to 128 so that slow extended precision REAL variables did not have to be used (section 5.2.7.1). Each sample size was simulated with a different set of frequencies of disease in the population. Samples with 8 cases were simulated with a frequency of disease of 1/2, samples with 16 cases with frequencies of

disease of 1/2 and 1/4, and samples with 32 cases, or more, were simulated with frequencies of 1/2, 1/4 and 1/8. For each sample size, at each frequency of disease, ROC curves with 4, 8 and 16 points were simulated. Each test was run 1,000 times to produce one chi-squared result. Even this very limited experiment took 24 days on a 1.4 GHz Athlon PC.

6.5.1 Nonparametric AUC Results

The results are given in appendix Table B.5. Given that the number of simulations was severely restricted by the computer time required to generate each pdf, all that can be strictly concluded is that the Monte Carlo simulation failed to prove that the method was not working. None of the chi-squared results is higher than 35, and only one is higher than 30. The chi-squared results from all 36 experiments have been pooled and are shown in Figure 6.8. This is not a particularly good practice in general, since errors at a particular combination of sample size, frequency of disease and number of points could be masked by correct results at other combinations, but with only one result from each experiment the situation doesn't arise. The combined histogram is what should be expected from a working experiment. This is therefore an encouraging preliminary result, but it would be rash to claim more.

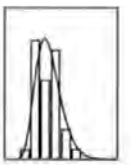


Figure 6.8 Combined histogram for nonparametric AUC

6.6 Parametric ROC Parameters and AUC

The Monte Carlo testing of the algorithm for parametric ROC curves is not as straight forward as other simulations in this chapter.

As discussed in section 5.5, the parametric ROC curve is fully specified by two ROC points. Third and subsequent points either lie on the curve, or the curve is nonparametric.

The algorithm to calculate the pdf in $\Delta\mu'$, σ_h space also calculates the total probability of these nonparametric curves.

It has been argued by the Gedankenexperiment in section 6.1.3 that Monte Carlo simulations only work for low sample size if the population point is uniformly distributed in probability space. The probability space of a three point ROC curve is six dimensional, but the parametric pdf is only four dimensional. The volume of this four dimensional hyper-surface is infinitesimal compared with the volume of the six dimensional hyper-cube, and hence the parametric pdf is an infinitesimal fraction of the nonparametric pdf.

However, the algorithm deals with quantized probability space, and therefore the parametric surface has a finite thickness of one element, and therefore a finite volume. However, the total probability of the nonparametric curves is still likely to be orders of magnitude larger than the total probability of the parametric curves (the sum of all the $\Delta\mu'$, σ_h elements) which therefore may make it difficult to detect errors in the posterior intervals caused by erroneous calculation of the parametric distribution. Therefore, the number of tests per chi-square result was increased to detect small errors in the pdf. To test if this larger number of tests was capable of detecting errors in the parametric pdf, the parametric pdf was deliberately bastardised and the chi-square value of the bastardised versions was taken. The parametric part of the pdf was multiplied by 0.5. 0.8, 1.25, and 2.0, and combined with the unaltered nonparametric part. The correct pdf is equivalent to multiplying by 1.0, and so 5 pdfs were generated for each test. Any errors in the chi-squared distributions of these bastardised pdfs (0.5, 0.8, 1.25, 2.0) therefore show that the numbers used for each chi-squared test are adequate for detecting errors, and therefore that correct chi-squared results for the unbastardised (1.0) pdf indicates it is actually correct.

Three Monte Carlo simulations were therefore run. The first simulation generated 2 point ROC curves to test if the one-to-one mapping from probability space to $\Delta\mu'$, σ_h space, and then from $\Delta\mu'$, σ_h space to AUC space, produces the correct pdfs in the transformed coordinates. The simulation used the largest grid size available on a 32 bit PC with the parametric grid set to 129 × 129, and the AUC grid to 129. Population points were randomly sampled from uniform probability space, and transformed into $\Delta\mu'$, σ_h, and

AUC space. Samples with 2^n, n=0..7 cases were simulated. Each test was run 2,000 times, and the experiment repeated 50 times to obtain a histogram of the chi-squared values of the parameters, and the AUC. The simulation ran for 165 days on a 1.4 GHz Athlon PC.

The second experiment generated 3 point ROC curves, and used bastardised parametric pdfs to detect if the parametric pdf was both detectable and correct. Preliminary tests indicated that more tests would be required the higher the sample size, and that at high sample sizes chi-square tests would need 20,000 tests using a parametric grid size if 65 × 65. (Detectability gets worse the larger the grid size so a smaller grid size helps). The parametric grid size was therefore set to 65 × 65, with an AUC grid size of 65, and samples with 2^n, n=0..7 cases were simulated 2,000, 2,000, 2,000, 2,000, 5,000, 10,000, 20,000, and 20,000 times respectively. The experiment was repeated 50 times to obtain a histogram of the chi-squared values, and ran for 33 days on a 0.8 GHz Pentium III PC.

The third experiment generated 4 point ROC curves. Sample sizes of 2^n, n=0..6 cases were simulated 2,000, 2,000, 2,000, 2,000, 5,000, 10,000, 20,000, and 20,000 times respectively on a grid size of 65 × 65. Preliminary tests indicated that by sample sizes of 128 cases the parametric pdf would be undetectable so the sample size only went up to 64. The experiment was repeated 50 times to obtain a histogram of the chi-squared values, and ran for 45 days on a 0.8 GHz Pentium III PC.

6.6.1 Parametric ROC Parameter and AUC Results

The results of the two point ROC curve test is given in appendix Table B.6 for the parameters, and Table B.7 for the AUC. Histograms for half of these results are given in Figure 6.9. The AUC results show the expected chi-squared distributions, however the parametric results are shifted towards chi-squared values that are too high. Given correct results for one mapping, and incorrect results for another mapping of the same experiment, this suggests either a error in the code for the parametric mapping or perhaps a subtle effect due to the limited grid size and granularity of the ROC curves. Very few lines define each ROC curve, but many ROC curves will have the same AUC, and therefore more lines define an AUC than define a ROC curve, which may have this effect.

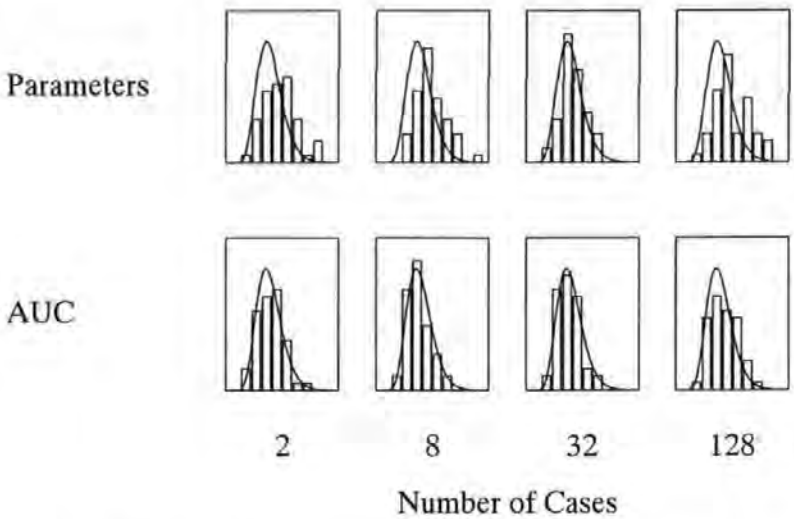


Figure 6.9 Histograms for 2 point ROC curve parameters

The results of the three point ROC curve test is given in appendix Tables B.8 to B.12 for the parameter test, and B.13 to B.17 for the AUC test. The effect of bastardising the parametric part of the pdf can clearly be seen for sample sizes of 64 and 128 cases which use 20,000 tests for per chi–squared. This is shown in the histograms in Figure 6.10. Multiplying the parametric pdf by 0.5 or 2.0 causes an obvious incorrect result. Multiplying the pdf by 0.8 or 1.25 causes slight divergence from the correct result, while the correct pdf (multiplying by 1.0) gives a good result. This effect is not very noticeable for sample sizes of 1, 2, 4, 8 where only 2,000 tests have been used per chi–squared, except when multiplying the pdf by 2.0. The effect is a little more noticeable for 16 and 32 cases, with 5,000, and 10,000 tests per chi–squared respectively. These trends can be seen in both the parameter and AUC tests.

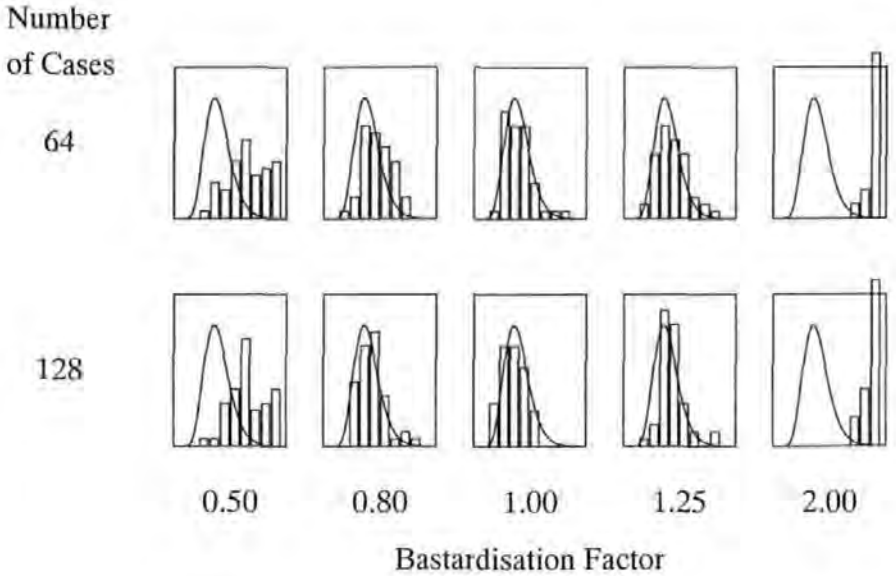


Figure 6.10 Histograms for 3 point ROC curve parameters

In contrast, the four point ROC curve test results given in appendix Tables B.18 to B.27 totally fail to show the detection of the correct parametric pdf within the nonparametric pdf. All tests appear to show the correct chi-squared distribution even when the parametric pdf has been deliberately bastardised. Even summing of the results for 16 and 32 cases into one chi-squared results of 1,000,000 runs failed to detect the deliberate errors. Given the parametric pdf is a 4 dimensional volume in a 8 dimensional space, this is not that surprising. It will probably require Monte Carlo simulations with many million of tests per chi-squared to detect the effect, which will have a huge run-time.

It should be noted that even detecting deliberately incorrect the point ROC curve pdfs is not enough to prove the distribution is correct. Even the smallest distortion of multiplying the pdf by 0.8 is a gross distortion. Ideally distortions in the order of 0.99 (or 1.01) should be investigated. However, that would require huge numbers of simulations. For now it should only be concluded that these experiments have failed to prove the parametric method wrong.

6.7 Weighted Confusion Matrix

The grid size was set to 257 × 257. Samples with 2^n, n=0..7 cases were simulated. For each sample size, at each frequency of disease, weighted confusion matrices with 2 (the

unweighted confusion matrix), 3, 5, 9 and 17 weights were simulated. One set of weights was generated and the test was run 2,000 times to generate each chi-squared value. The procedure was repeated 40 times.

Preliminary tests suggested that the algorithm is sensitive to increasing sample size and increasing numbers of weights, probably because the weights do not quantize well. While increasing the grid size can compensate, run time is proportional to the product of the probability grid size and the weight grid size ($G_x \times G_w$). It was therefore decided to limit the sample size to 128, which had the added advantage that all the calculations could be done in fast double precision arithmetic rather than slower REAL arithmetic (see section 5.2.7.1). As it is, the 40 chi-squared tests that were run took 96 days on a 0.8 GHz Pentium III PC.

6.7.1 Weighted Confusion Matrix Results

The results of the Monte Carlo simulation are given in appendix Table B.28. Histograms of six results for a five weight confusion matrices are plotted in Figure 6.11. As illustrated by the diagrams, the experimental results show the expected chi-squared distributions. Given the number of cases simulated, 80,000 in each histogram, this indicates that the method is working within the limitations of the quantization of the weighted confusion matrix into 257 elements, and the stochastic nature of Monte Carlo simulations.

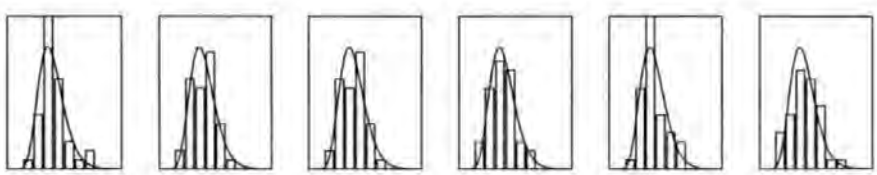


Figure 6.11 Histograms of confusion matrix with 5 weights

6.8 Summary

This chapter has presented the Monte Carlo simulations used to test all the novel methods introduced in Chapter 5. Only pilot studies were run for some methods because of the com-

puting resources required. With one exception, all the methods gave good results. The parametric test for a two point ROC curve gave correct results for the AUC but slightly incorrect results for the parameters. This anomaly needs investigation. The fast methods for nonparametric ROC points, and for nonparametric ROC point comparisons, have been tested so thoroughly there can be no doubt about their validity. The confusion matrix test ran for a sufficient number of iterations that there can be little doubt that the method is correct. The method for the nonparametric AUC is so slow that the limited number of tests that were run only failed to prove the method is invalid.

This chapter also presented a Bayesian argument that the population values used to generate the samples have to be drawn from a uniform random distribution. All the Monte Carlo experiments use this method, and all produce the correct result. When a Frequentist fixed population point was tested the experiment failed as predicted. Many Monte Carlo experiments reported in the literature use a Frequentist fixed population point, which explains why they are unsatisfactory at low sample size.

Using a uniformly distributed random population point presented great difficulty for the parametric ROC curve test, since the parametric pdf is an infinitesimal fraction of nonparametric space when the curve has more than two points. A bastardised parametric pdf was therefore used to determine if the Monte Carlo simulations were sensitive enough to detect errors in the parametric pdf.

7 Applications

The methods described in Chapter 5 were applied to two examples of ROC analysis, published in the literature, in order to investigate the posterior boundaries produced on real data, and to illustrate how the proposed method could enhance the analysis.

The confusion matrices analysis was applied to a weighted confusion matrix first used to illustrate weighted kappa by Cohen [14].

7.1 ROC Examples

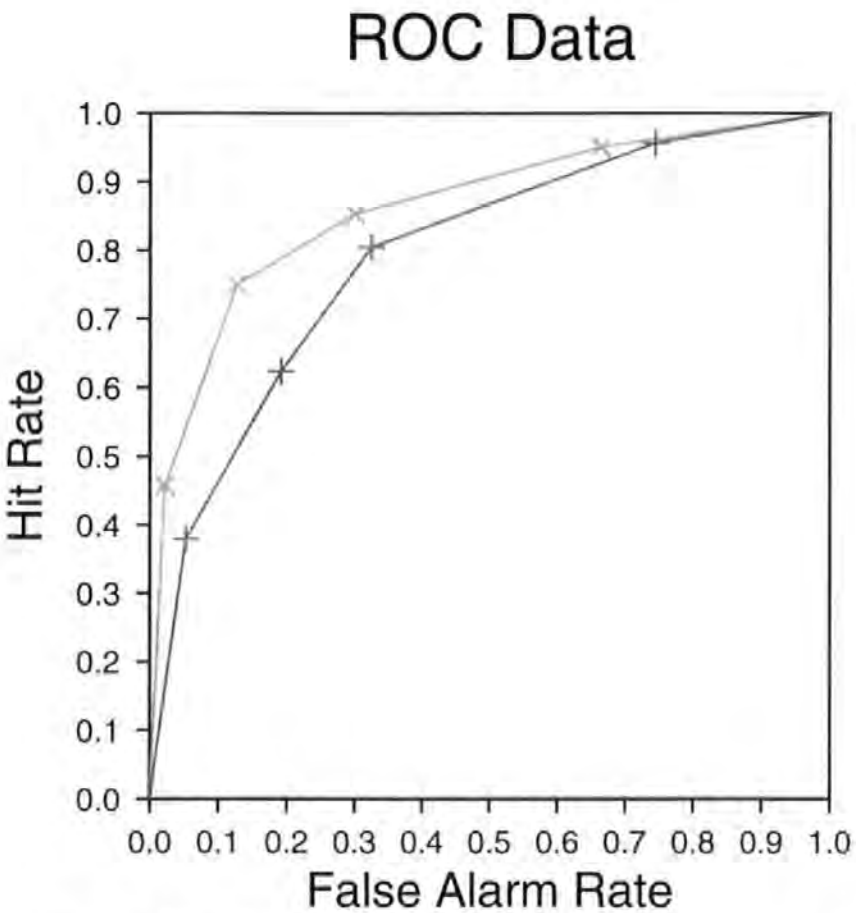


Figure 7.1 ROC curve of diagnosis of 708 mammograms [21]

The first example is taken from Swets [21]. Swets recommended the use of ROC analysis for measuring the accuracy of many types of diagnostic systems. A radiological example was presented to illustrate the use of ROC analysis, and the Area Under the Curve, as the preferred single-valued measure of accuracy. A study had previously been carried out,

in which six radiologists were asked to examine 118 mammograms (58 malignant, 60 benign), and classify them into one of five categories, according to likelihood that the lesion was malignant. The radiologists first diagnosed the mammograms unaided (denoted as 'standard'), and then used two diagnostic aids (denoted as 'enhanced'). The raw data for the pooled categorisations were given in the paper, allowing the ROC graph for the standard and enhanced diagnoses to be reproduced here as Figure 7.1.

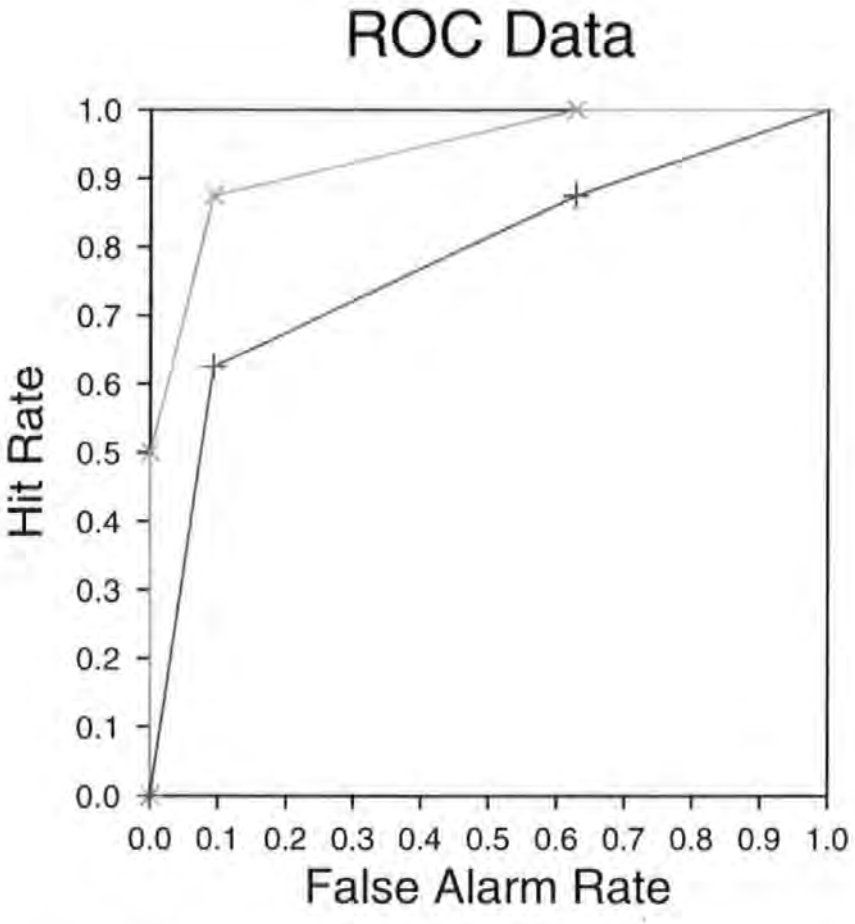


Figure 7.2 ROC curve of diagnosis of 51 pancreatic cases [10]

It should be noted that the validity of pooling the data from six experts to produce one ROC curve is debatable. However, the data does provide a useful example of a ROC curve with a high sample size.

The second example is taken from Adlassnig and Scheithauer [10], in which an expert system, known as CADIAG-2/PANCREAS, for the differential diagnosis of ten different types of pancreatic disease, is described. The performance of the system was compared

to an histologically or clinically confirmed 'Gold–Standard' diagnosis. There were 47 patient records available in which one or more of a subset of six pancreatic diseases had been diagnosed. Four patients had dual diagnoses, giving a total of 51 diagnoses of one of six diseases. A series of ROC graphs were presented, illustrating the performance of the CADIAG–2 system in the differential diagnosis of specific diseases, both using what was described as a 'limited' set of patient data and with the 'full' set of available patient data. Two of Adlassnig & Scheithauer's ROC curves, their Figures 9 and 10, illustrate the evaluations of 8 diagnoses of acute pancreatitis from the 51 cases compared to the 'Gold–Standard', using 'limited' patient data and 'full' patient data respectively. Although the raw data were not given, they can be reconstructed from the ROC graphs, because the number of cases was small. The data are combined here and reproduced as Figure 7.2. However, originally 11 categories were used, which because of the small number of cases often left zero cases in a category which gives an ambiguous reconstruction. Here five categories have been reconstructed. It should be noted that this makes the posterior intervals larger because the order term is smaller (assuming a uniform Bayesian prior), so this re–analysis is actually slightly unfair to Adlassnig & Scheithauer, but still illustrates the point about sample sizes.

7.2 Nonparametric ROC Points

From the raw data the 95% posterior boundary of each point on the four curves was calculated by the program described in Section 5.2.7 using a grid of 512 × 512, and these posterior boundaries are shown in Figures 7.3, and 7.4. The entire process of calculating the ROC probability density functions, and determining the 95% boundaries took approximately 2.2, and 2.6 seconds respectively on a 1.4 GHz Athlon PC.

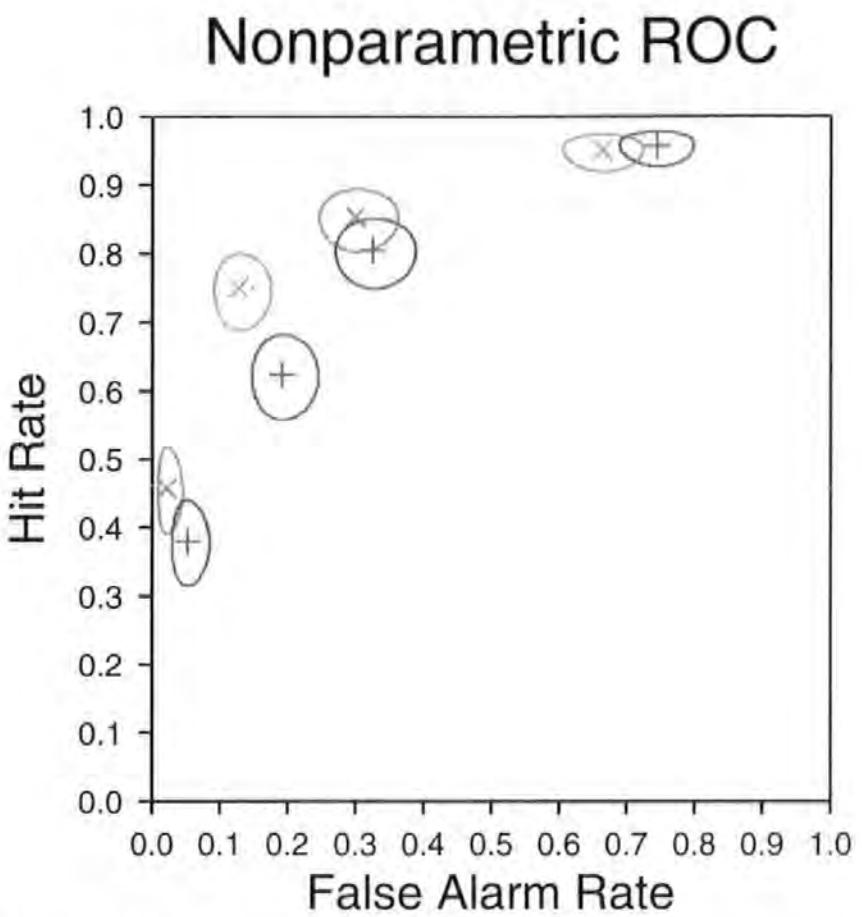


Figure 7.3 Posterior boundaries of ROC points of mammogram data

From Figure 7.4, it is obvious that there is considerable uncertainty in the results due to the limited number of cases used. In their analysis, Adlassnig and Scheithauer state that accuracy was always increased by adding the 'full' patient data, in accordance with anticipation. While the ROC curves presented in Figure 7.2 appear to support this common-sense conclusion, the large posterior boundaries in Figure 7.4 suggest that this conclusion was probably premature given the data.

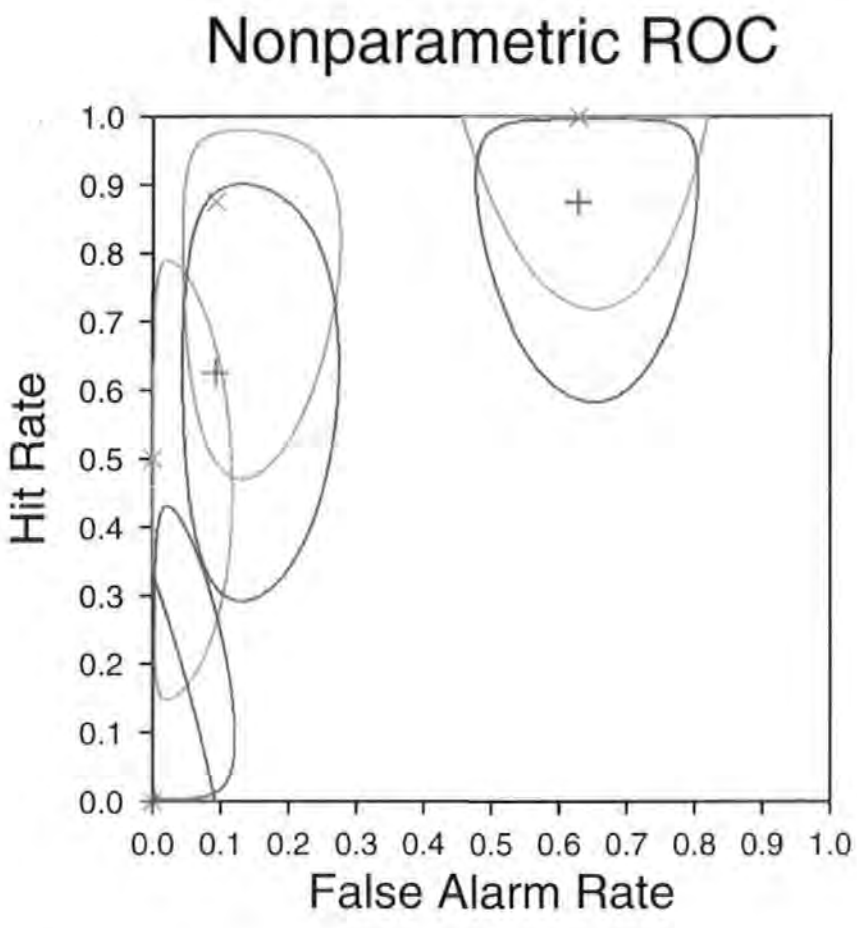


Figure 7.4 Posterior boundaries of ROC points of pancreatic data

Figures 7.3 and 7.4 clearly illustrate the difference that sample size makes to the confidence that can be placed in the location of each point. While the ROC curve of the mammogram diagnoses in Figure 7.1 do not look as accurate as the ROC curve for the diagnosis of acute pancreatitis in Figure 7.2, examination of the posterior boundaries in Figures 7.3 and 7.4 shows that the 708 (six opinions of 118) mammograph cases are sufficient to give good confidence of the location of the ROC points, while the 51 pancreatic cases give a much larger posterior boundary.

In particular, it can be seen by consideration of pairwise points in Figure 7.3, that the points are outside of each others posterior boundaries in all cases, and that the posterior boundaries are mutually exclusive in one case. In contrast, in Figure 7.4, there is a high degree of overlap in posterior boundaries in all cases. In particular, the points with False Alarm Rate of 0.093 lie within each others posterior boundaries, and the 'limited' data

point with False Alarm Rate of 0.638 lies well within the posterior boundary of the 'full' point.

7.3 Nonparametric ROC Point Comparison

This difference between points can be quantified by the method described in section 5.3.1. Figure 7.5 shows the difference between each point given by the 'enhanced' and 'standard' mammogram data from Swets. Figure 7.6 shows the difference between each point given by the 'full' and 'limited' acute pancreatitus data from Adlassnig and Scheithauer. The format of the graph is described in section 5.3.1.2.

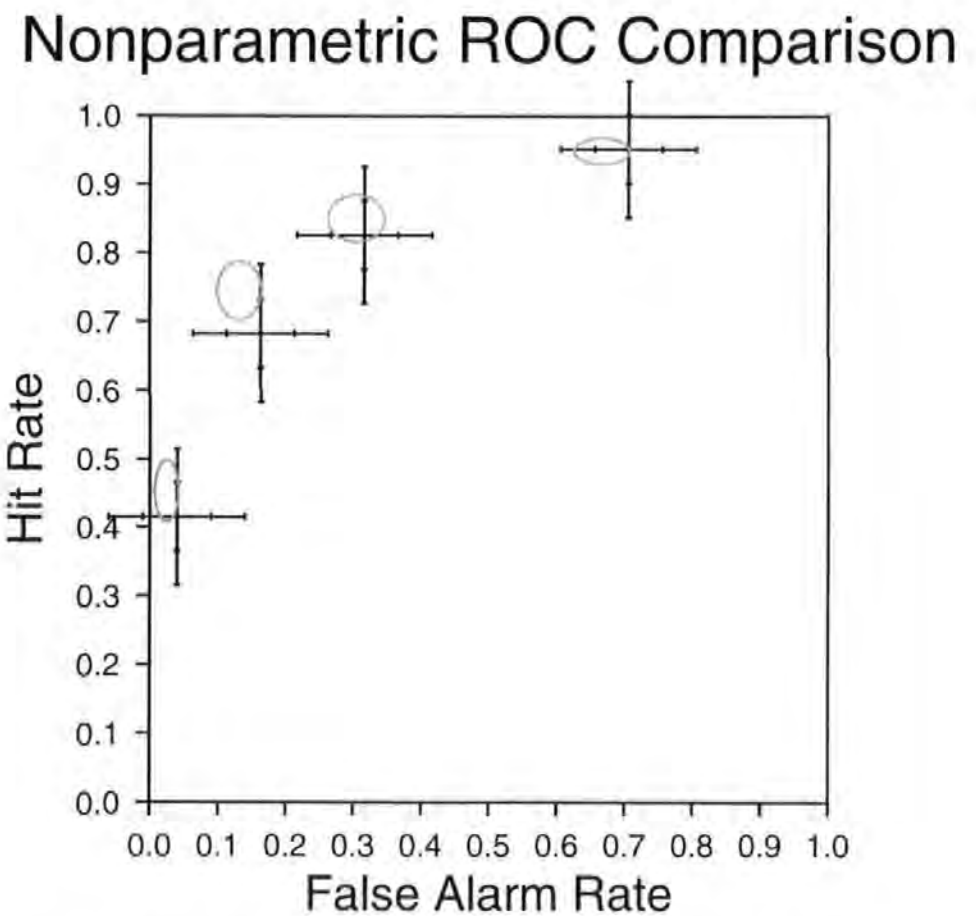


Figure 7.5 Posterior boundaries of difference of ROC points in 7.3

These graphs were plotted using a comparison grid size of 1023 × 1023 (equivalent to a ROC grid size of 512 × 512) and took 5.6 and 6 seconds respectively to calculate on a 1.4 GHz Athlon PC.

More detail is provided by Figures 7.7 and 7.8. Figure 7.7 shows the difference between the third points of Swets' data. Figure 7.8 shows the difference between the first points of Adlassnig and Scheithauer's data. Section 5.3.1.1 describes the format of these graphs in detail. Again a grid size of 1023 × 1023 was used, and the program took 5.6 and 6 seconds respectively. The program produced a graph of this type for every point, only two of which have been included for brevity.

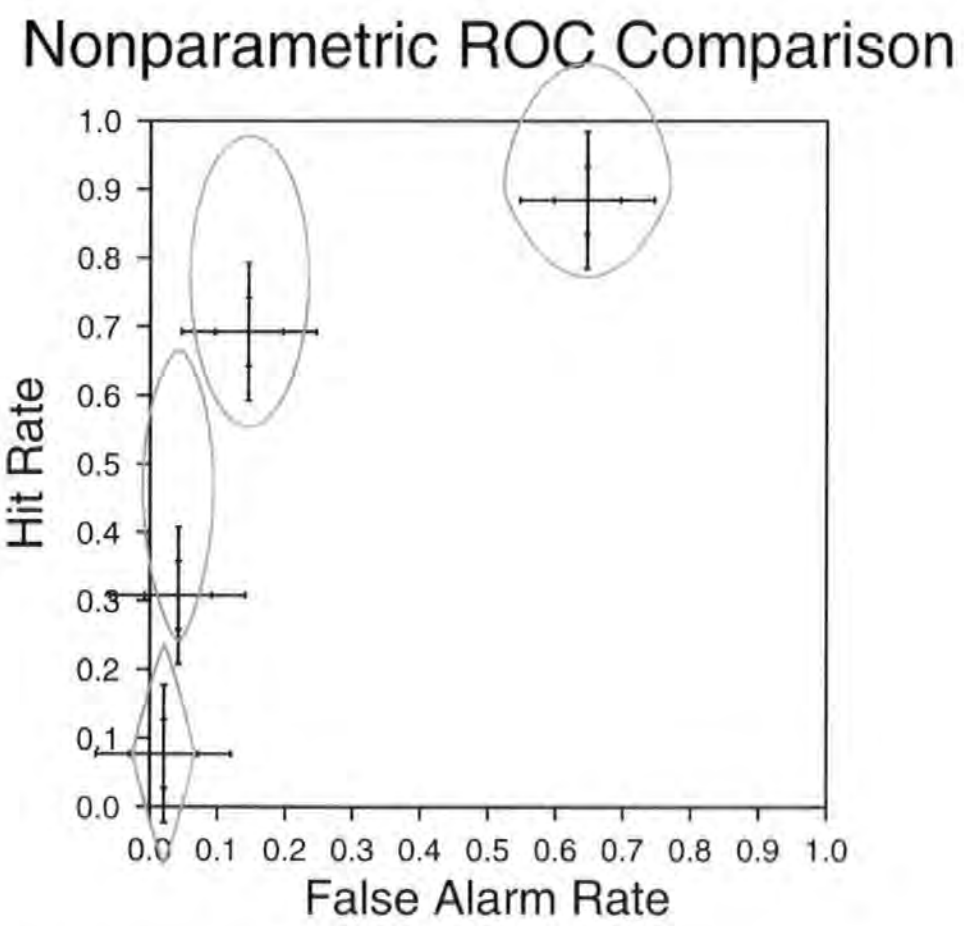


Figure 7.6 Posterior boundaries of difference of ROC points in 7.4

These comparisons bear out the observations from the plots of the ROC point posterior intervals. In the mammogram ROC curves the 95% posterior intervals of the differences between the first two points are almost entirely within the upper left quadrant meaning that there is a high probability (96.4% and 99% respectively) that the 'enhanced' test has both a higher Hit Rate and lower False Alarm Rate at this point. At the third point (Figure 7.7) the certainty is not as high, being only 72.9%, but there is a high probability of 95.5%

that the 'enhanced' test does have a higher Hit Rate. At the fourth point it is doubtful if the 'enhanced' test gives a higher Hit Rate, but there is a 99.1% probability that it gives a lower False Alarm Rate.

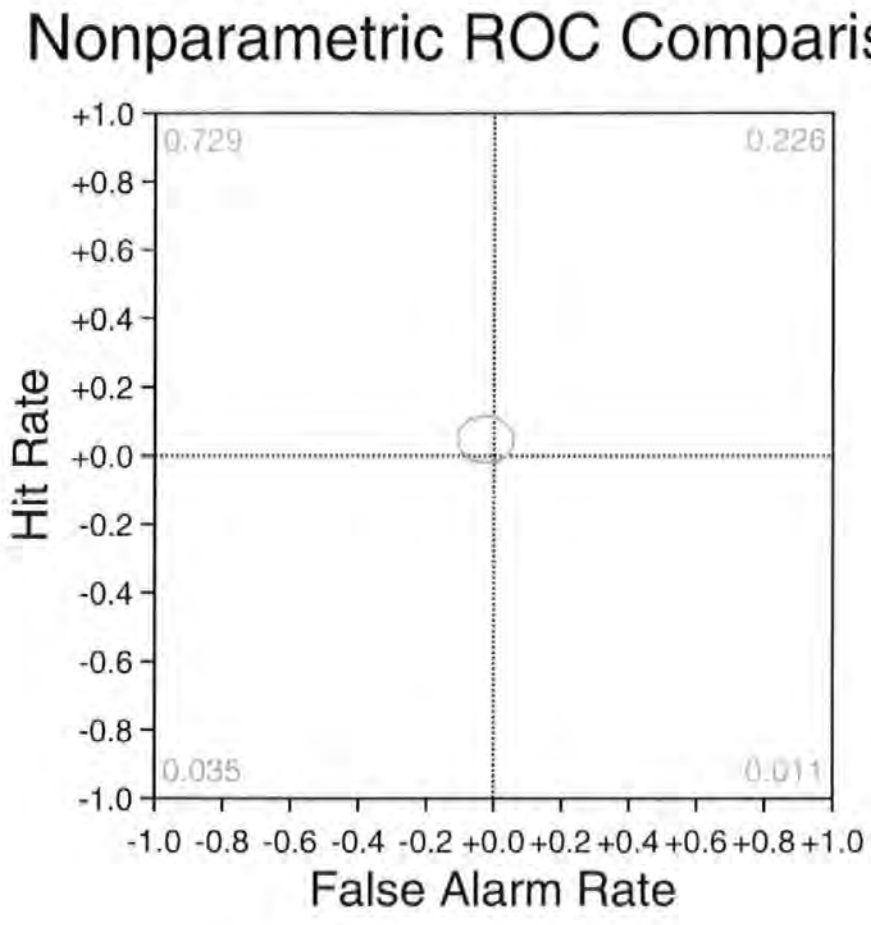


Figure 7.7 Detail of the difference of 3rd points in 7.5

The acute pancreatitis comparison (Figures 7.6 and 7.8) just reinforce how little faith can be given to the apparently large differences in the location of ROC points when the sample size is low. The first point (Figure 7.8) also shows an unexpected shape for the posterior interval. This illustrates the benefit of going back to basic principles and following the logic through, rather than making dubious assumption about the form of posterior/confidence intervals as has been described elsewhere in the literatrure (see section 3.2.7), though when the sample size is high, i.e. the 708 cases of mammogram data, the posterior intervals appear eliptical (Figures 7.3 and 7.5).

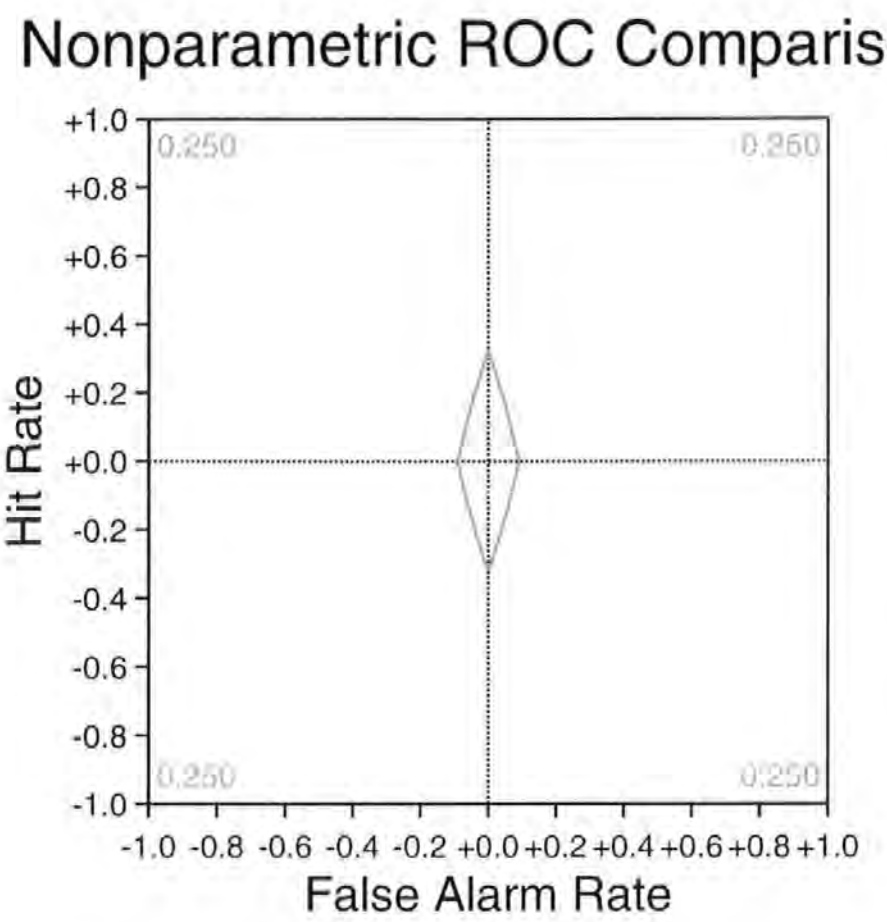


Figure 7.8 Detail of the difference of 1st point in 7.6

7.4 The Nonparametric AUC

Although Adlassnig and Scheithauer [10] described the use of the AUC, including testing AUC differences statistically in order to compare ROC curves, no results were given for the curves obtained.

Swets used a parametric method to estimate a 'maximum–likelihood' curve through the points, and a corresponding parametric estimate of the AUC and its standard error. The values Swets obtained for the parametric AUC were 0.81 and 0.87, with standard errors of 0.017 and 0.014 for the 'standard' and 'enhanced' diagnoses respectively. Parametrically, the 95% confidence interval is given by the mean value ± 1.96 × standard error, which gives a 95% confidence interval of 0.777 to 0.843 for the 'standard' data and 0.843 to 0.897 for the 'enhanced' data.

The nonparametric method for calculating the AUC described in section 5.4.1 was used to plot the pdf of the AUC, and calculate the 95% posterior interval, for both Sweets' (Figure 7.9) and Adlassnig and Scheithauer's (Figure 7.10) data. The grid size was set to 257, and the program took over 4.3 hours on a 1.4 GHz Athlon PC to produce each graph.

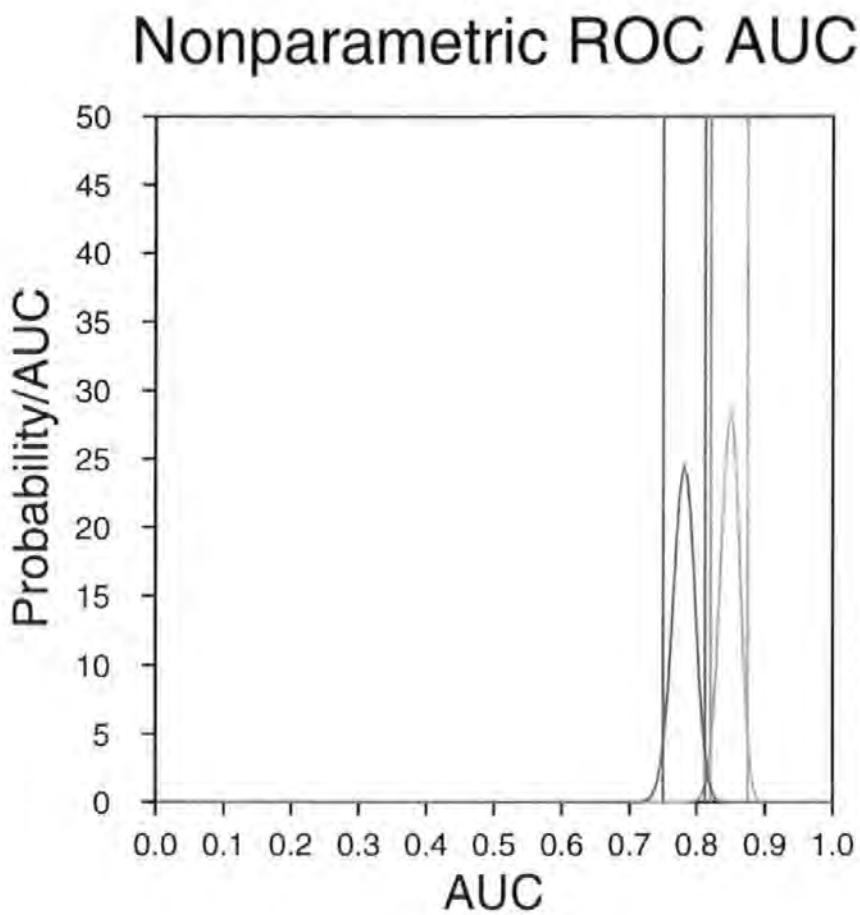


Figure 7.9 Nonparametric AUC of mammogram data

Using this nonparametric Bayesian method the 95% posterior interval of the AUC of the 'enhanced' curve was 0.82 to 0.875 and the 'standard' curve 0.75 to 0.8125, which is similar to the result obtained by Sweets for the Frequentist parametric confidence interval. Setting aside any differences between the Bayesian and Frequentist approaches, the difference can be explained because the AUC of a convex curve is smaller when calculated with the trapezoid rule than when calculated by fitting a smooth curve (section 1.1.4).

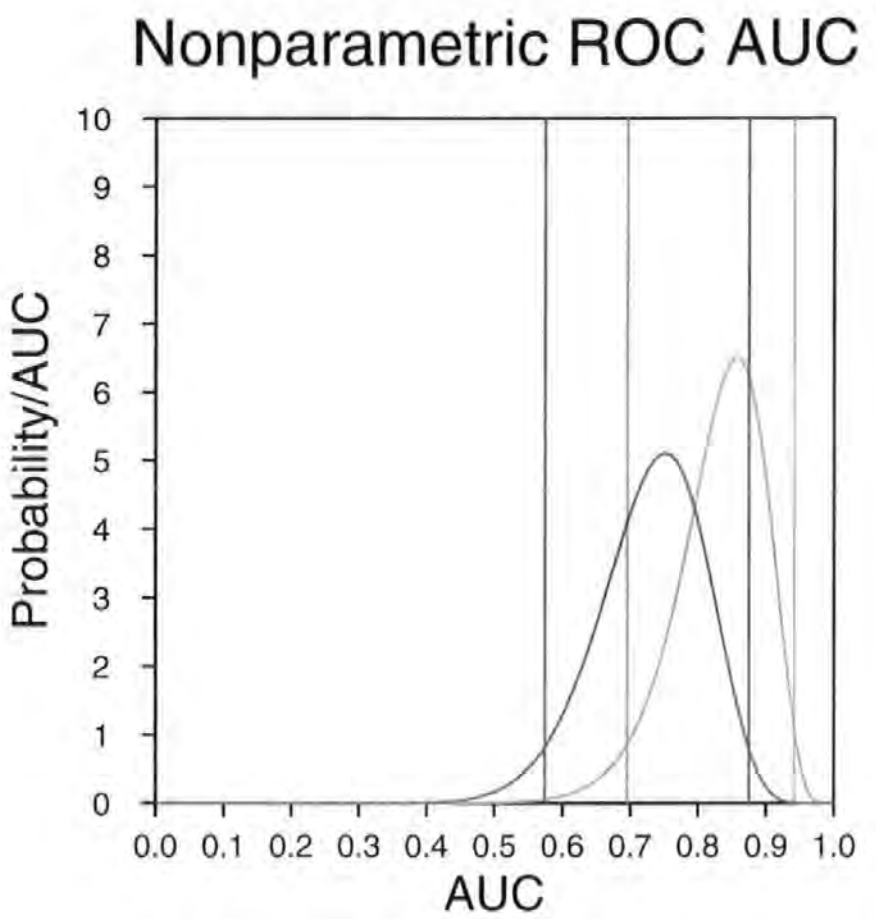


Figure 7.10 Nonparametric AUC of pancreatic data

The acute pancreatitis data gives a 95% posterior interval for the nonparametric AUC of 0.69 to 0.94 for the 'full' data, and 0.56 to 0.87 for the 'limited' data. Again, the overlap between the posterior intervals shows that any claim that the 'full' data shows improvement over the 'limited' data is weak.

7.5 The Parametric AUC

The parametric confidence intervals of the AUC can be obtained by using the method in section 5.5.1. This allows a direct comparison with the figures Swets quoted for the Frequentist confidence interval. Figure 7.11 gives the pdf of the AUC calculated from the mammogram data, and Figure 7.12 gives the pdf of the AUC of the acute pancreatitus data. These graphs were produced using a grid size of 129 and took 4.5 minutes on a 1.4 GHz Athlon PC.

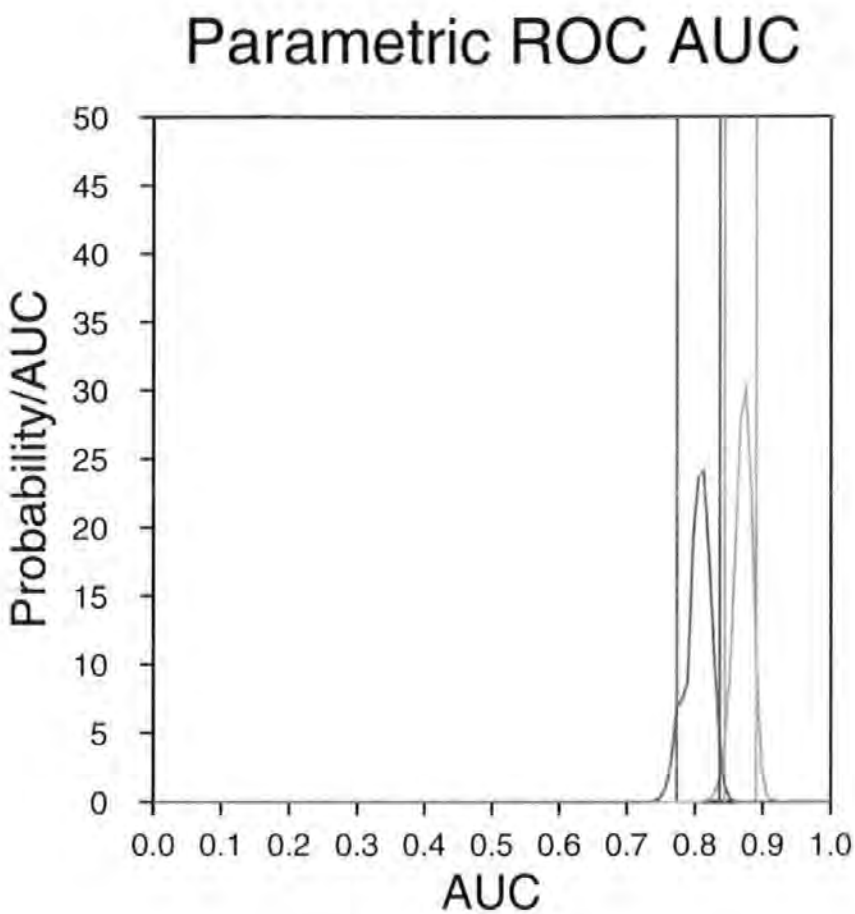


Figure 7.11 Parametric AUC of mammogram data

The Bayesian posterior interval for the 'enhanced' mammogram parametric AUC was 0.84 to 0.89, and the 'standard' 0.77 to 0.84, which is very close to the figures Sweets gave for the Frequentist confidence interval (0.843 to 0.897 and 0.777 to 0.843). Given that the figures calculated here are quantized to fractions of 1/128, they give an excellent match within the accuracy of the calculation, which indicates that the Bayesian method actually gives the same answer as existing Frequentist parametric methods for large sample sizes (in this instance 708). This was also shown by the Monte Carlo experiment using a Frequentist fixed population point (section 6.3).

The posterior interval of the 'full' acute pancreatitus AUC was 0.75 to 0.97, and the 'limited' 0.58 to 0.88. As no figures were given by Adlassnig and Scheithauer, a comparison cannot be made.

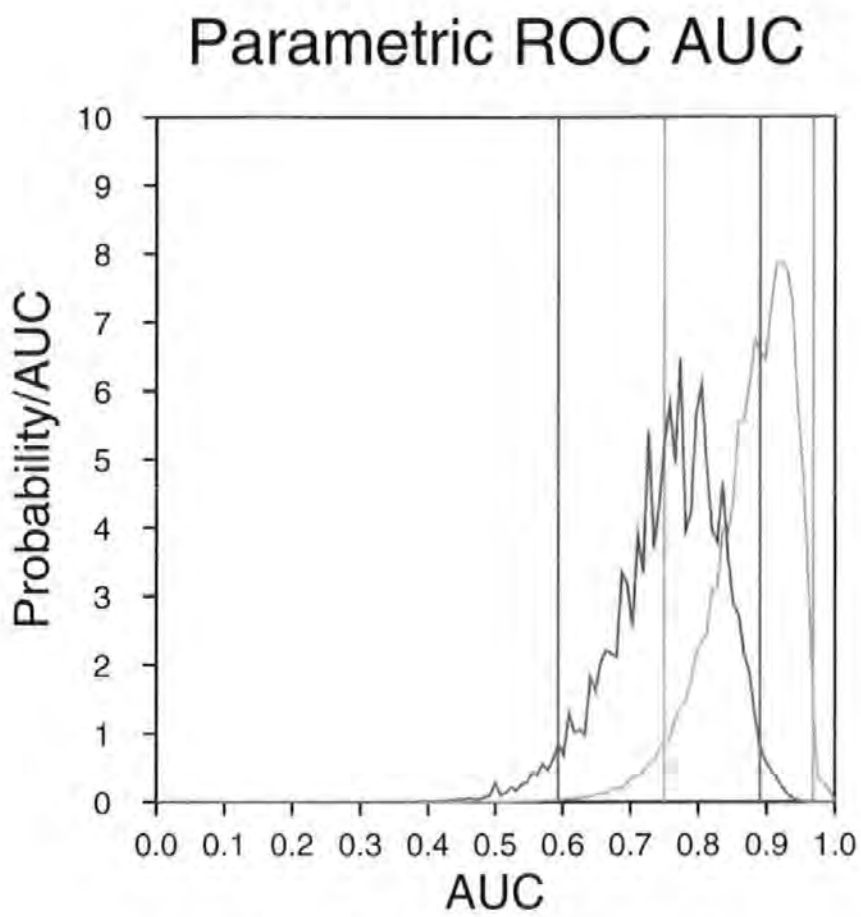


Figure 7.12 Parametric AUC of pancreatic data

It is interesting to note the differences in these figures from the nonparametric AUC for the same data. For the 'limited' set the nonparametric posterior interval is slightly smaller as would be expected by the trapezoid rule giving a smaller AUC for a convex curve, but the discrepancy with the 'full' data is much wider. The upper limit is 0.94 compared with 0.97, which can possibly be explained by the trapezoid rule, but the lower limit is 0.69 compared with 0.75. This might be due to the trapezoid rule, or might be something more fundimental. The noisiness of the graph should also be noted.

7.6 Parameter Plots

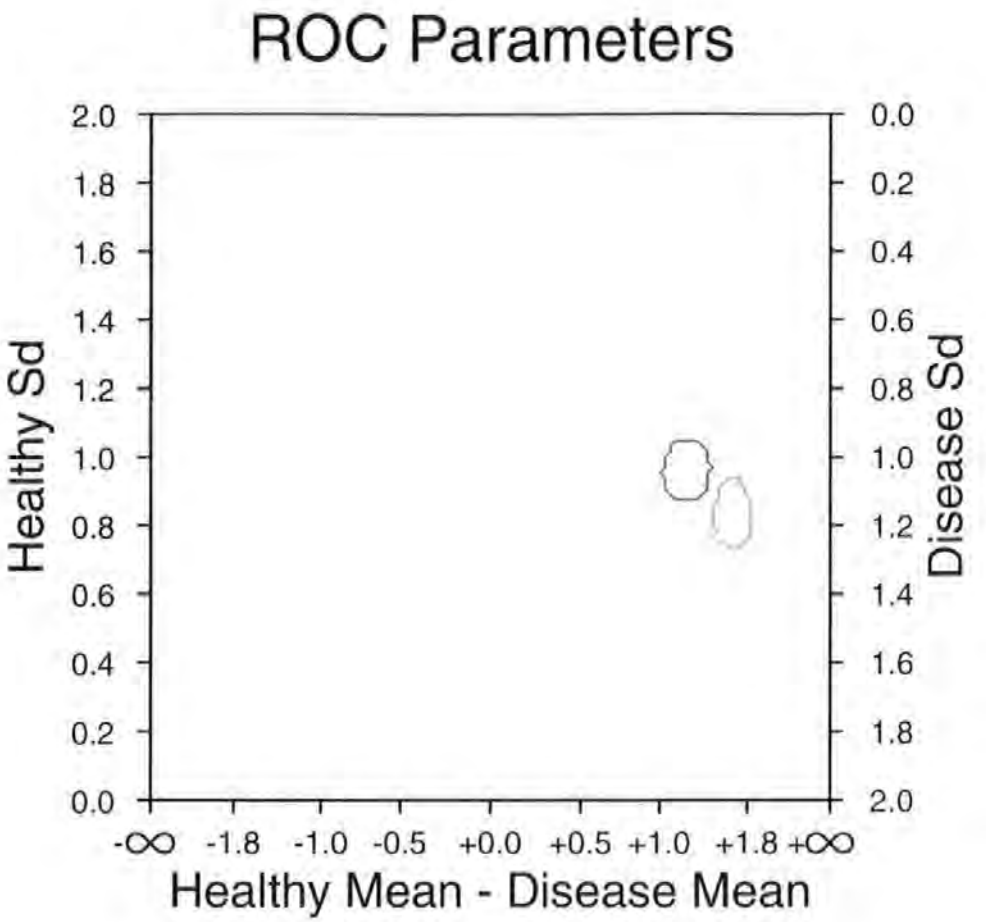


Figure 7.13 Parameters of mammogram data

The same program that produces the parametric AUC also produced a plot of the parameters of the ROC curve. Figure 7.13 shows the 95% posterior intervals of the parameters of Swets' data, while Figure 7.14 shows the posterior interval for Adlassnig and Scheithauer's 'full' data, and Figure 7.15 the 'limited' data. These graphs have been plotted seperately because both are noisy and difficult to distinguish when plotted on the same graph.

The noise is probably due to the small grid size (129) and to the data sets. The first two categories of the 'limited' data set are zero. In the normal terms of maximum likelihood analysis this data set would be classified as 'degenerate', and will not give any result in a standard maximum likelihood fit. Here, it has been observed that such data sets fragment the pdf, but do not break the program. Adlassnig and Scheithauer's data would appear

to be rather an extreme case, which is not helped by the low grid size. It should be noted that the noise does not occur in Figure 7.13 (or Figure 7.11), which indicates it is data dependant.

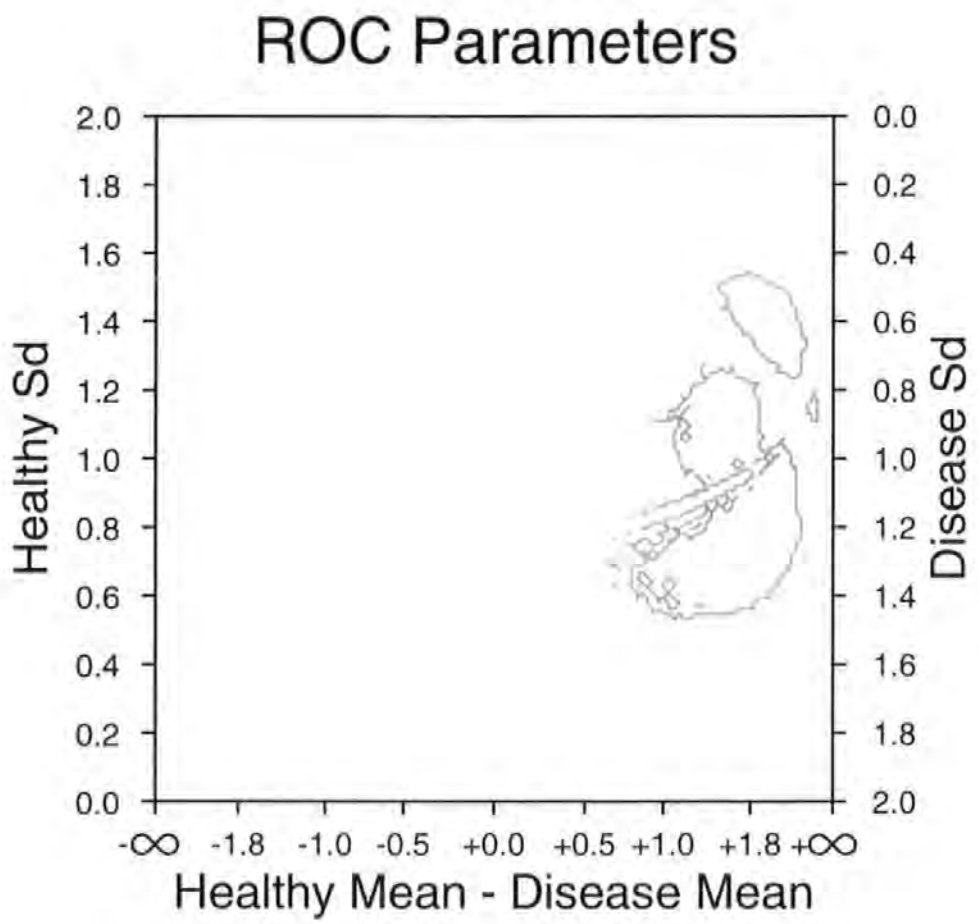


Figure 7.14 Parameters of 'full' pancreatic data

This type of plot is entirely novel. It delivers a previously unseen representation of parametric ROC curves, and makes explicit parameters that have only been discussed in abstract terms before.

These plot gives an alternative way of visualising a ROC curve. A binormal ROC curve is completely determined by the two parameters $\Delta\mu'$, σ_h, and therefore a single point on this plot completely specifies a ROC curve. This parsimonious representation allow several ROC curves to be compared, and their features clearly seen, as can be observed with the two distinct confidence intervals of the mammogram ROC curves in Figure 7.13. In certain circumstances this can give far more information than existing methods. The pos-

terior limits around these points allows ROC curves to be compared. Having two parameters presents more information than the AUC, and localising the posterior interval around a point is clearer than presenting posterior bounds on a ROC graph where several curves are being compared.

The $\Delta\mu'$, σ_h plot may also provide an alternative, and powerful way, of rigorously specifying the performance requirement of an intelligent medical system in terms of a region of the plot. The posterior interval obtained from a system evaluation can be used to calculate the probability that the system is within the clinically useful range.

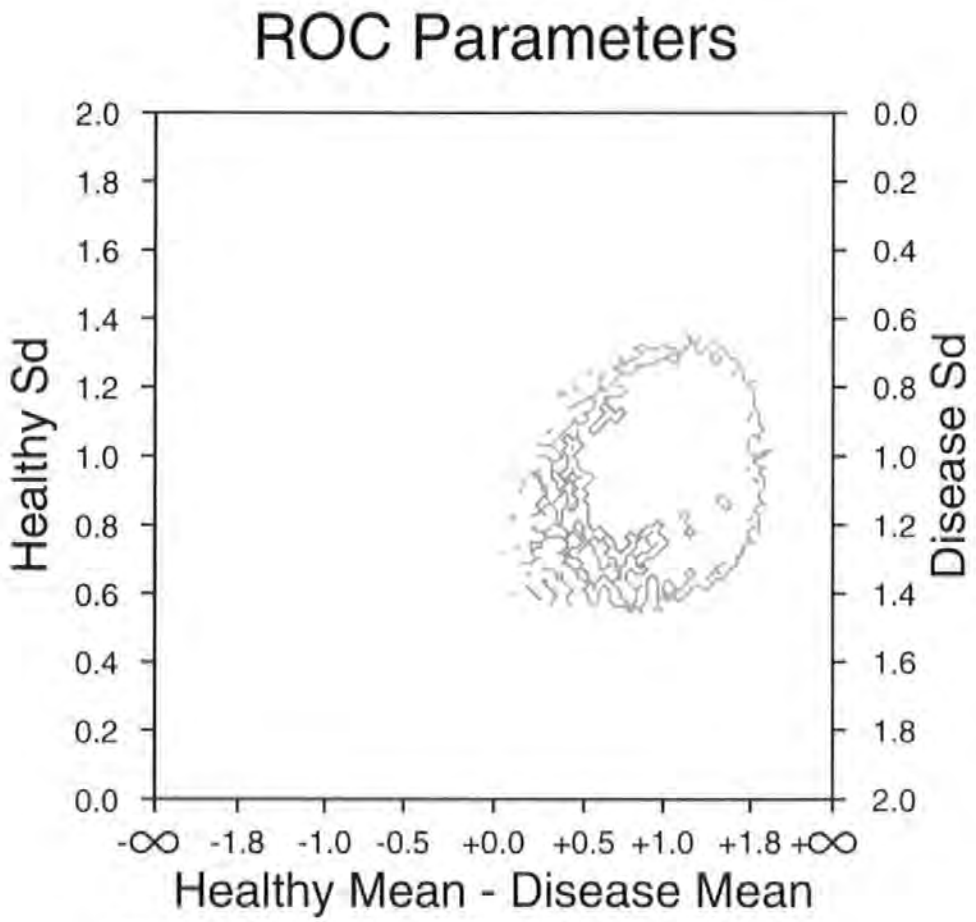


Figure 7.15 Parameters of 'limited' pancreatic data

7.7 Maximum Likelihood Curves

The parametric analysis can also be used to produce a maximum likelihood curve. Figure 7.16 gives the maximum likelihood curves for Swets' 'enhanced' and 'standard' data, while Figure 7.17, gives the maximum likelihood curves for Adlassnig and Scheithauer's

'full' and 'limited' data. There are three essential differences between these plots and existing methods of maximum likelihood analysis.

Firstly, the parameters are quantized, which gives a discrete set of possible ROC curves from which the one of highest probability is picked. The correspondence with the true maximum likelihood curve will be imprecise at low grid size, and since memory space is proportional to the fourth power of the grid size, this is a real limitation in practice.

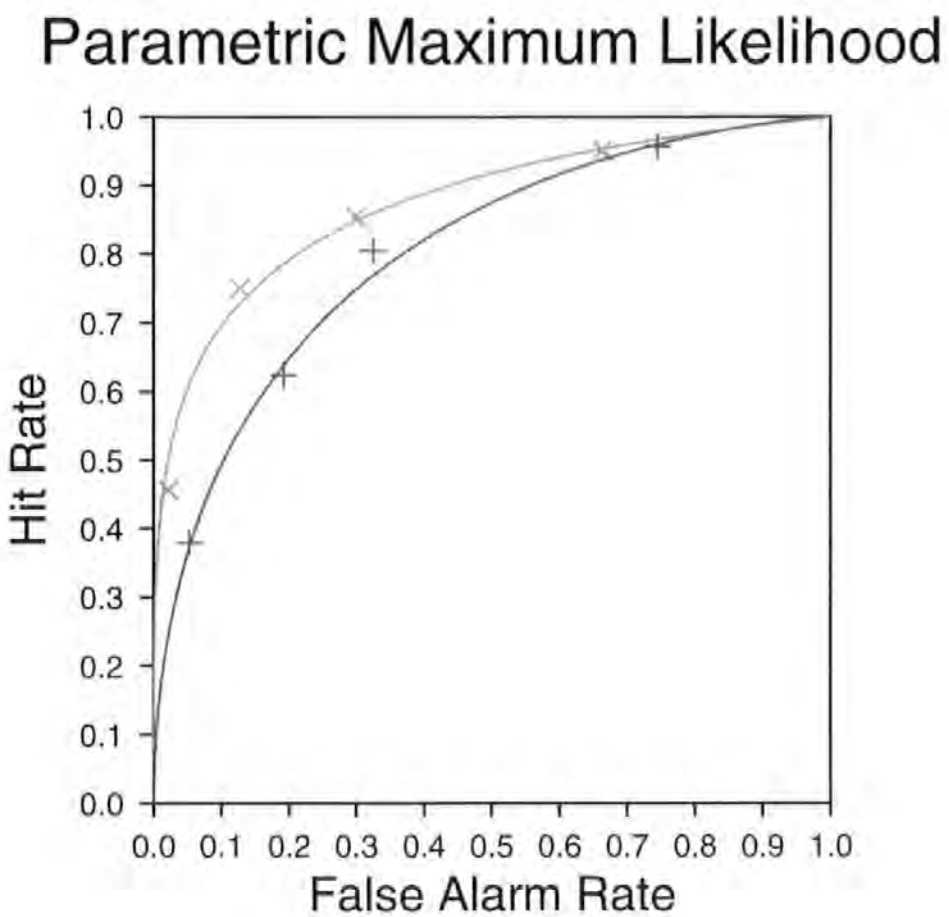


Figure 7.16 Maximum likelihood curve of mammogram data

Secondly, the maximum likelihood curve is plotted using the parameter values where the pdf is at a maximum on the parameter plot (e.g. Figures 7.13, 7.14 and 7.15). This point

is the sum of the probability of all possible thresholds values that have that pair of parameters. This is in complete contrast to existing maximum likelihood methods which search for the point of maximum probability of a given combination of the parameters and thresholds. In the absence of categories with zero cases this will be the same point, but where there are zero cases, it will be different. This leads on to the final point.

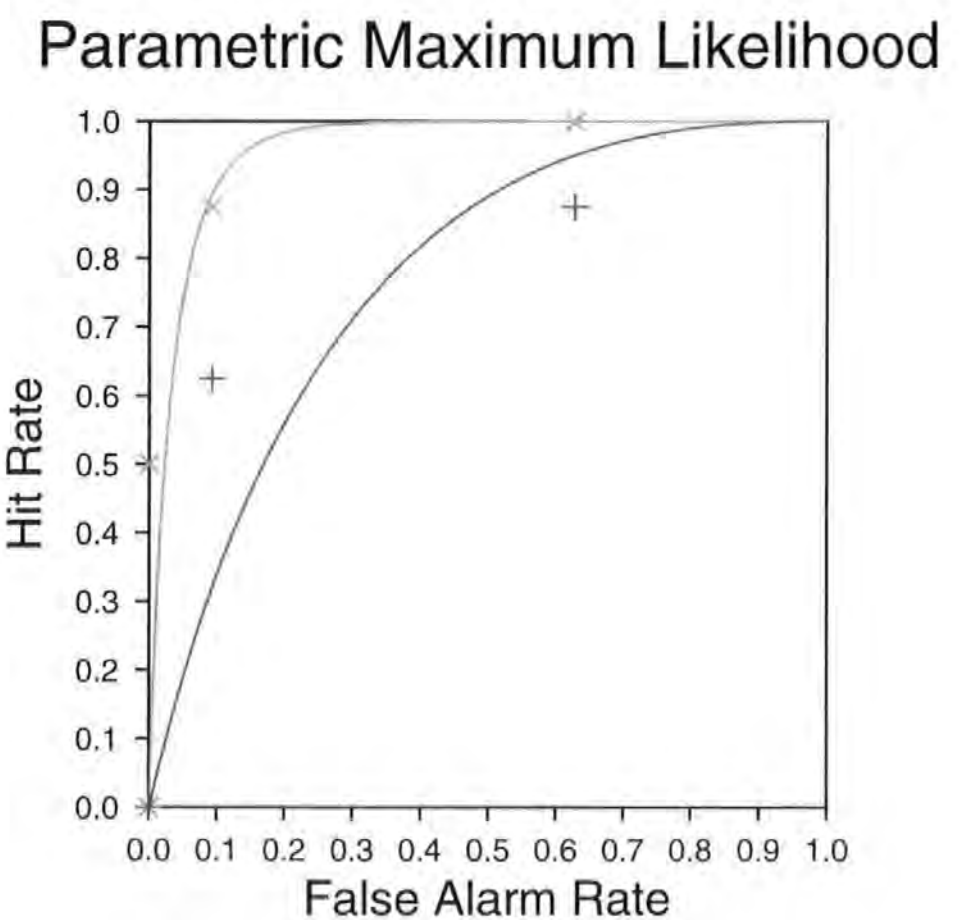


Figure 7.17 Maximum likelihood curve of pancreatic data

Thirdly, this method of maximum likelihood is robust. It can deal with 'degenerate' data sets without breaking. Given that small data sets are the most likely to be 'degenerate'

it gives robustness where it is needed. Achieving robustness at small sample sizes is an aim of this research.

7.8 Confusion Matrix Example

Finally, the plot of a weighted confusion matrix will be demonstrated. The example is taken from Cohen's original paper on weighted kappa [14]. Figure 7.18 reproduces the values in Cohen's example (Category A was Personality disorder, B was Neurosis, and C was Psychosis), and includes the weights in parenthesis. It should be noted that here agreement is scored 0, and disagreements as 1, 3 or 6.

		'Gold Standard'			
		A	B	C	Row Sum
Test	A	88 (0)	14 (1)	18 (3)	120
	B	10 (1)	40 (0)	10 (6)	60
	C	2 (3)	6 (6)	12 (0)	20
Column Sum		100	60	60	200

Figure 7.18 Example weighted confusion matrix from Cohen

The weighted kappa value is 0.348 with a 95% confidence interval of 0.171 to 0.525 (in a kappa range of −1 to 1).

Figure 7.19 shows the pdf as calculated by the algorithm in section 5.6.2. The 95% posterior interval is 0.70 to 1.17 (in a weight range of 0 to 6).

These two intervals bear little resemblance to each other because they measure different things. The confusion matrix posterior interval gives the bounds on the weight that can be expected if this system (or human expert) was used for diagnosis in the population. The usefulness of this depends on the basis on which the weights were assigned to the matrix. If these are carefully considered clinical measurements of the consequences of each (mis)classification in the matrix, for instance, suicide rate per thousand per year, useful inferences can be drawn. In this example there is a 95% chance the suicide rate of this expert's patients will be between 0.7 and 1.17 per thousand per year. On the other hand,

if this is some arbitrary guess at the misery of the patients, such statistical precision is hardly warranted.

This contrasts with the kappa approach where the confidence interval can be interpreted to mean [76] the expert has achieved between a 'slight' and 'moderate' agreement with the 'Gold Standard' diagnosis (see section 4.3).

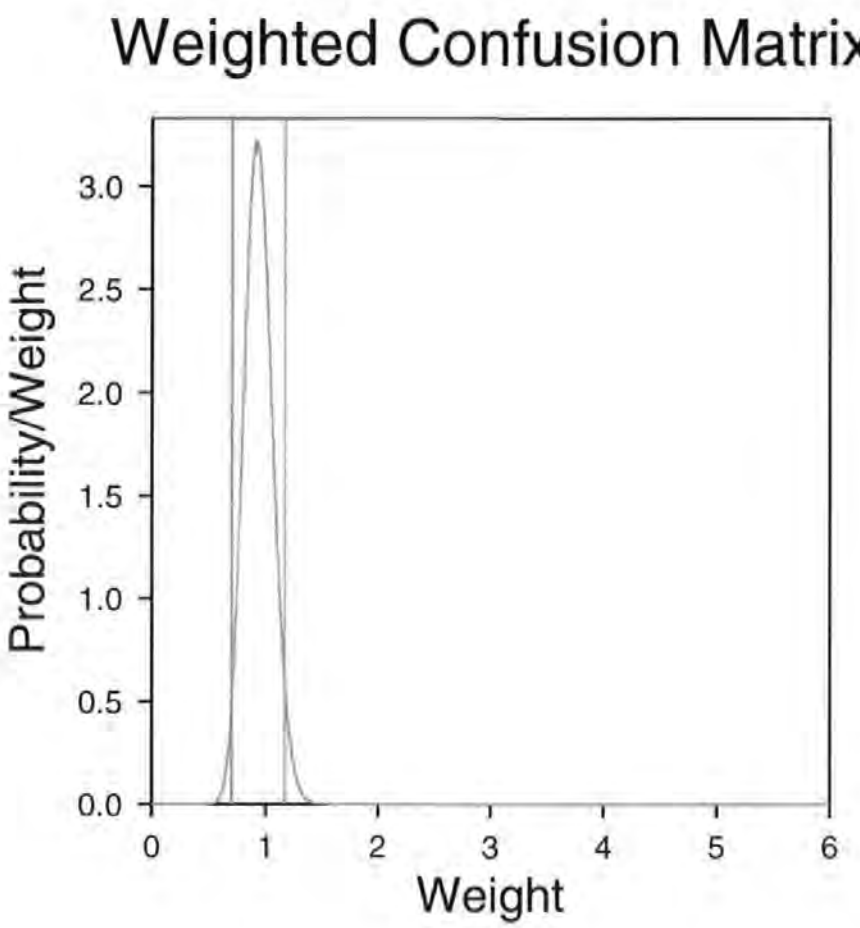


Figure 7.19 Weight pdf from Cohen's example

More utility might be achieved in comparing two experts. Cohen [14] states that the kappa statistics are Gaussian at large sample size. For now it will be assumed that 200 cases is sufficient for this approximation to hold true. If the kappa statistic and standard error are calculated for two experts the probability that one expert is better than another could be calculated from the two Gaussian distributions. (c.f. section 5.3). However, this gives a pdf in terms of the difference in kappa, which presumably could only be given as a linguis-

tic description of the difference in diagnostic performance. Calculating the pdf of the difference in weight is potentially more useful.

7.9 Summary

This chapter has presented demonstrations of all the novel methods introduced in Chapter 5. Two sets of ROC data from the literature, one with 708 cases, the other with 51 cases, were used to demonstrate the ROC methods. The importance of using posterior intervals when drawing inferences from ROC curves produced with small samples has been highlighted. Perhaps more importantly, it has also demonstrated that at high sample sizes the novel methods give the same intervals as existing methods, even when one is a Frequentist confidence interval, and another is a Bayesian posterior interval, and that at low sample sizes the distributions are far from Gaussian.

The utility of the weighted confusion matrix posterior interval in producing an objective measure of expected system performance has been explained. This is in contrast to kappa statistics which can only give a measure of agreement.

8 Discussion and Conclusion

8.1 Discussion

If an intelligent medical system is to be deployed it needs to be proved to be as good as the human experts it replaces or supplements. That proof can only be provided by testing the system on a representative sample of cases that human experts actually deal with. These test cases have to be collected, and given a 'Gold Standard' diagnosis. Then the performance of a group of experts has to be compared with that of the system. Human experts are rare and busy, which severely restricts the number of test cases that can be collected, and the number of test cases that can be used in a comparative study. The statistical analysis of the results of these comparative studies therefore have to be accurate for small sample sizes.

ROC curves and confusion matrices are ideal representations for analysing intelligent system evaluation results. Unfortunately, existing methods of analysis do not work at small sample sizes. It is not even clear at what sample size they are valid.

This thesis has therefore focused on presented statistical methods for ROC curves and confusion matrices that are robust and accurate for all sample sizes, which will therefore be important for intelligent system evaluations conducted with small sample sizes.

ROC curves can either be analysed using parametric or nonparametric statistics. Parametric analysis introduces assumptions about the underlying form of the populations which cannot be verified from the data, though it could be said that nonparametric analysis makes the assumption of straight line interpolation between known points on the distributions. Parametric analysis commonly makes the assumption that the distributions of the populations of healthy and diseased cases are Gaussian (normal) which gives a 'binormal' ROC curve. Both methods can be used to provide the Area Under the Curve (AUC) which gives an overall measure of the accuracy of the system under test. However, it may be clinically more relevant to focus on certain parts of the curve. All methods seem popular in the literature, and so this thesis has presented accurate statistics for them all.

Section 5.2 presented a method for calculating the nonparametric posterior bound of each point of a ROC curve. It extends the work of Hilgers [42] in using the beta function to calculate the pdf of the location of the population ROC point from a sample. Hilgers obtained separate pdfs for the Hit Rate and False Alarm Rate and combined them to give a rectangular posterior boundary for a single point. Section 5.1 shows how to calculate the exact posterior boundary shape, rather than a crude rectangle, while section 5.2 goes on to extend the method to multiple points. This analysis introduced the idea that the pdf of a ROC curve should be seen as a multidimensional object (Dirichlet distribution), of which the various posterior bounds are two dimensional projections. This analysis produced an order term, dependant on the Bayesian prior (section 5.2.5). This rigorous mathematical analysis has also been extensively validated by a Monte Carlo simulation of 818,400,000 experiments. The method produced robust and accurate posterior bounds but also uncovers a problem in deciding a priori how many categories to use when collecting ROC data under the assumption of a uniform Bayesian prior (section 5.2.6).

Section 5.4 presented an algorithm for calculating the pdf of the nonparametric AUC. The algorithm relies on factorising the expression for the probability of a point in the multi-dimensional pdf of the ROC curve, and thus the algorithm runs in time proportional to the number of ROC points. It is unfortunate that calculating the AUC of each point requires a calculation to the fifth power of the grid size of the quantized AUC pdf. Due to the slow speed of the algorithm, only 36,000 Monte Carlo experiments were run to validate the method, which is too few to conclusively prove the method is valid. However, using Popperian philosophy, it can be said the experiments failed to show the method was invalid.

The factorising algorithm is of general utility. Section 5.5 shows how it can also be used to generate the pdf of the parameters, and AUC, of a binormal parametric ROC curve. The pdf can be used to produce a maximum likelihood curve for the data points. This has the advantage over existing methods of being robust for all data sets. The plot of the posterior bound of the parameters is entirely novel in ROC analysis. The disadvantage of the method is the computational resources required. The algorithm requires memory proportional to the fourth power of the grid size, which is a limitation with a 32 bit memory ad-

dress. The Monte Carlo testing gave correct results for the AUC, but incorrect results for the parameters, which requires further investigation.

Confusion matrices can either be weighted or unweighted (actually binary weighted). The analysis of an unweighted confusion matrix is identical to the analysis of the False Alarm Rate (or equivalently the Hit Rate) of a ROC curve, and so the same methods can be applied. The factorising algorithm can again be harnessed to provide an algorithm for calculating the pdf of the weights of a weighted confusion matrix (section 5.6.2). While Monte Carlo simulations indicate the method works well for small sample sizes and a modest number of different weights, preliminary investigations indicated that the algorithm was sensitive to large sample sizes and large numbers of weights which then require a large grid with a finer mesh size to compensate. It was thus too computationally expensive to run Monte Carlo simulations for large sample sizes and numbers of weights.

The pdf of the weight of a weighted confusion matrix provides a different measure from weighted kappa. Providing clinically meaningful weights can be assigned, the method can provide the posterior interval of the weight of the system. The weights might be QALYs, micromorts, or pounds Sterling. In contrast the kappa statistic only provides a measure of agreement on an interval scale that can then be given a linguistic interpretation. It does not appear that kappa can be used to provide exact figures to calculate the risks and benefits of deploying a system in practice. However, kappa does have the advantage of being chance corrected. A method for chance correcting the confusion matrix pdf is discussed below (section 8.3.5).

In order to compare system against expert, methods are required to compare the pdfs produced by the various methods above. Section 5.3 discusses a method of comparing the pdfs of each point on a ROC curve. Again, extensive Monte Carlo experiments validated that the method does work as expected. It is robust and accurate even in situations that would not be expected in practice.

The algorithms to generate all the above posterior boundaries have been implemented in a suite of programs to make them available to intelligent system evaluators and other inter-

ested parties. They provide an alternative to existing methods. Appendix C documents the available software in more detail.

Despite Fisher[37][38], and Neyman[39] using the beta distribution for distribution free tolerance regions as long ago as the 1930s, it does not appear to have had the exposure it merits. Green and Swets[89] (according to Metz [57]) suggest the following formula for the standard deviation of the Hit Rate (and an equivalent formula for the False Alarm Rate), which can be used to plot crossed error bars around individual ROC points:

$$\sigma = \sqrt{\frac{Hit\ Rate \times (1 - Hit\ Rate)}{No.\ True\ positives + No.\ False\ negatives - 1}}$$

According to this equation two true positives and zero false negatives would give a standard deviation of zero, which implies the pdf of the sample Hit Rate is a zero width spike at a Hit Rate of one. While this is a rigorous result within the Frequentist paradigm, it is counter intuitive in the common (mis)understanding that a Frequentist confidence interval is the same as a Bayesian posterior interval. A textbook [90], gives the same equation but does state the conditions for use of the equation. These conditions exclude the example above, and recommend Bayesian statistical tables which give the 95% and 99% Bayesian posterior intervals of the beta function [91].

For large sample sizes, where the numbers of false positives is similar to the number of true positives, the method does tend towards identical results to Bayesian methods (the beta distribution asymptotically approaches the Gaussian distribution for large sample sizes).

8.1.1 Limitations of the Mann–Whitney Test

The Mann–Whitney U statistic is the basis of methods for analysing the nonparametric AUC of a ROC curve. It is commonly found in textbooks, but can only be used for hypothesis testing. Hypothesis testing weighs the probability of a set of propositions. For instance, hypothesis 0 may state that the population AUC of a given ROC curve is 0.5, and hypothesis 1 may state that the AUC is 0.3. Mann and Whitney calculated the distribution of the sample given an AUC of 0.5, so the probability that a given sample is generated by an

AUC of 0.5 is known. Assuming the same can be done for the an AUC of 0.3, these probabilities can be compared. Given that the AUC of 0.5 gives a (binomial) distribution of possible samples and the AUC of 0.3 also gives a (binomial) distribution of possible samples, the situation could be modelled by a ROC curve. However, instead of finding an optimum point on this ROC curve, only the 0.5 AUC distribution is considered in determining a cut–off point, which is usually the 95% confidence interval. If the sample has less than a 5% chance of originating from an AUC of 0.5, hypothesis 0 is rejected. Since only hypothesis 0 is being considered, hypothesis 1 can become an 'alternative' hypothesis that the AUC is not 0.5, and the same logic is applied. In contrast, the Bayesian approach calculates the probability that the sample was produced by an AUC of 0.49, and then 0.48, and so on for every AUC from 0 to 1 to see which range of AUC is most likely, as described in section 5.4. It therefore provides an alternative to the Mann–Whitney U statistic where a Bayesian prior distribution can be agreed.

DeLong's method does produce the standard error of the population AUC for any sample. This statistic assumes the distribution of the population AUC is Gaussian. However, there is no definitive rule to determine the accuracy of the approximation given the sample size and sample AUC. Given that the method presented in section 5.4 is correct, it will calculate the exact Bayesian posterior pdf, and hence the exact posterior intervals, provided the user has access to a few hours computer time on a powerful PC.

The existing method of maximum likelihood analysis of binormal ROC curves has one major flaw – it falls over when presented with 'degenerate' data sets, which occur in increasing frequency the smaller the sample size and the higher the AUC. These are precisely the two conditions likely in intelligent medical system evaluations where a (hopefully) accurate system may have to be tested with an (unavoidably) small sample. The method also assumes the distribution of the population is Gaussian when the sample size is large, but the error in the approximation at low sample sizes has not been quantified. From section 7.5 it would appear that by a sample size of 708 (with an AUC of about 0.8) the maximum likelihood method gives the same answer as the new method presented in section 5.5, and it can be tentatively concluded the distribution is Gaussian. However, with a sample size

of 51, and an AUC of about 0.7, Figure 7.12 suggests the Bayesian posterior distribution is far from Gaussian (even allowing for suitable smoothing of this noisy curve). Somewhere between these two points the Gaussian distribution becomes an acceptable approximation.

The precise nature of all these approximations can be investigated with Monte Carlo experiments. The experiments of Obuchowski and Lieber ([19] and section 3.6.1) typify this approach. It is assumed that at some small sample size the Gaussian approximation will become inadequate, and that the results of the Monte Carlo experiments will show this divergence from correct behaviour as the sample size falls. The experiments do indeed show this. However, section 3.6.2 suggests this is an artifact of using a fixed population to generate the samples. Section 6.3 demonstrates this by re-running the working Monte Carlo experiment in section 6.2 using a fixed population point, and obtaining the predicted failure. The only surprise is the large sample size required before the experiment can be persuaded to work. It would therefore appear that the experiments reported in the literature are incapable of proving the hypothesis they set out to test. This is an inherent limitation of the Frequentist paradigm which rather handicaps any sensible discussion of the minimum sample size required to give good Gaussian approximation. In contrast, the Bayesian experiments reported in this thesis have used a uniformly distributed random population point, to match the uniform Bayesian prior distribution and have obtained good results from simulations using small sample sizes.

8.2 Contributions to Knowledge

To summarise, this thesis presents research that has achieved the following:

- Introducing accurately shaped two dimensional nonparametric confidence bounds to ROC curve points.
- Devised the correct pdf for the points of a multi-point nonparametric ROC, and hence discovering the expressions for multiple points include an order term when a uniform Bayesian prior is used.

- Produced an algorithm to calculate the exact posterior interval of the nonparametric AUC. If validated by further Monte Carlo simulations, it could provide an alternative method where researchers have access to good computer support.
- Shown an alternative way of analysing parametric ROC curves which solves the problem of degenerate data sets. While the Monte Carlo testing of the method gives good results for the AUC, there is a slight error with the parameters which needs further investigation.
- Provided an alternative way of looking at confusion matrices, though this work is incomplete as far as chance correction is concerned.

Finally, the methods in Chapter 5 may be specific instances of generic Bayesian solutions to low sample statistical problems. They may provide a paradigm for future research.

8.3 Future Work

There are three main areas for future work. Firstly, some of the Monte Carlo simulations reported in Chapter 6 should only be regarded as pilot studies even though they ran for weeks. These simulations should be run with much larger numbers of iterations to confirm the results of the pilot studies. This will take years on one or two PCs, so either a super-computer is required, or the simulations should be run on many PCs in parallel and the results combined.

Secondly, there may be better ways of generating the pdfs of the nonparametric AUC and the weight of a confusion matrix than the algorithms described in Chapter 5. Preliminary work on analytic solutions are presented below. Analysis of confusion matrices could be improved by including chance correction. A discussion of how this might be achieved is presented. It would also be desirable to extend the method of comparing the pdf of pairs of ROC points from two different systems to all the statistics presented in Chapter 5, and to account for correlation effects.

Thirdly, because this research has returned to basic principles to solve problems in a mature field, there are many aspects of ROC curve and confusion matrix analysis that have not been investigated. Re-analysis of these, based on the novel Bayesian methods described

here, may produce better solutions, particularly where small sample sizes are concerned. Possible avenues of research are discussed.

8.3.1 Further Monte Carlo Simulations

While the pilot study on the nonparametric AUC produced good results, only 1,000 iterations were run for each combination of sample size and number of ROC points in a test that took 24 days. It is suggested that at least 20 times this number of tests need to be run. This would take 16 months if run on the same PC that ran the pilot study, so clearly better computing resources are required.

The study on two point parametric ROC curves gave correct results for the AUC but slightly incorrect results for the parameters. The cause of this anomaly needs to be investigated. The tests on three point ROC curves were encouraging but the bastardisation factors were coarse. The test should be re–run with bastardisation factors closer to 1.0. Depending on the results, it may be worthwhile re–running the four point ROC curve tests with much larger numbers, though judging by the pilot study, even extensive runs on the most powerful supercomputer are unlikely to detect a correct parametric pdf against the much larger nonparametric pdf in realistic time.

8.3.2 Analytic Solution for Nonparametric AUC

Calculating the pdf of the nonparametric AUC is the most computationally expensive algorithm proposed in this thesis. Finding a better solution, preferably an analytic one, would be desirable.

The pdf of an n point ROC curve exists as a $2n$ dimensional hyperplane in a $2(n+1)$ dimensional hypercube. In order to generate the pdf of the Area Under the Curve all points in the pdf that correspond to the same AUC must be summed. In other words, the probability of a given AUC is the integral over the hyperplane pdf.

The AUC for a given set of points is given by the trapezoid rule.

Consider the first line segment of an n point ROC curve. This segment starts (by definition) at coordinate (0, 0), and ends at coordinate (x_0, y_0). The next segment start at (x_0, y_0) and ends at (x_1, y_1).

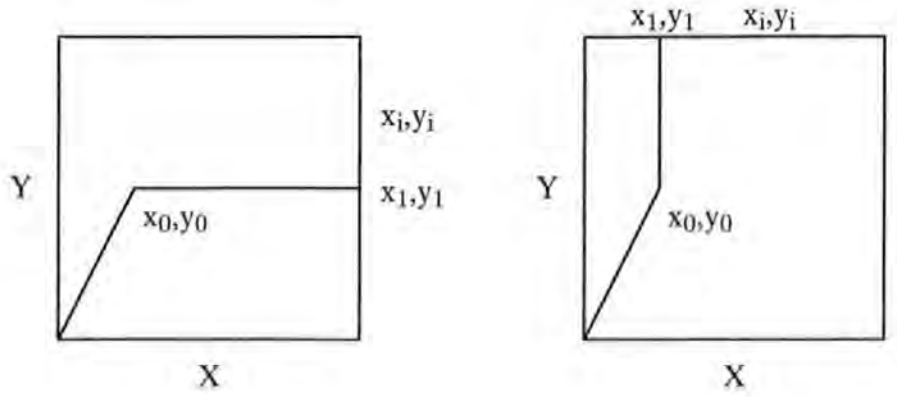


Figure 8.1 Minimum and maximum areas of first point

By definition, each x coordinate of a ROC point can never be less than the x coordinate of the preceding point. Similarly for y. Given an (x_0,y_0) coordinate the maximum AUC occurs when $x_1 = x_0$ and $y_1 = 1$, the minimum AUC occurs when $x_1 = 1$ and $y_1 = y_0$, as shown in Figure 8.1.

(This implies all other points x_i,y_i $(i = 2,3,...,n)$ on the curve follow the right boundary $(x_i = 1,\ y_i \geq y_1)$ in the case of a minimum, or the upper boundary $(x_i \geq x_1,\ y_i = 1)$ in the case of a maximum).

The minimum AUC is then:

$$MinAUC = \frac{x_0 y_0}{2} + (X - x_0)y_0$$

The maximum AUC is:

$$MaxAUC = \frac{x_0 y_0}{2} + (X - x_0)Y$$

Where X is the width of the ROC graph, and Y is the height. These are of course both 1 for the first point, but variables are used rather than constants because the general form of these equations will be used later.

The point x_0,y_0, can be at any location such that the total AUC is a. Thus two bounds can be drawn for the location of point x_0,y_0 corresponding to the lines:

$$x_0 = 2X - \frac{2a}{y_0}$$

$$x_0 = \frac{2a - 2XY}{y_0 - 2Y}$$

These boundaries are sketched in Figure 8.2 for values of a of 0.25, 0.50, and 0.75 XY.

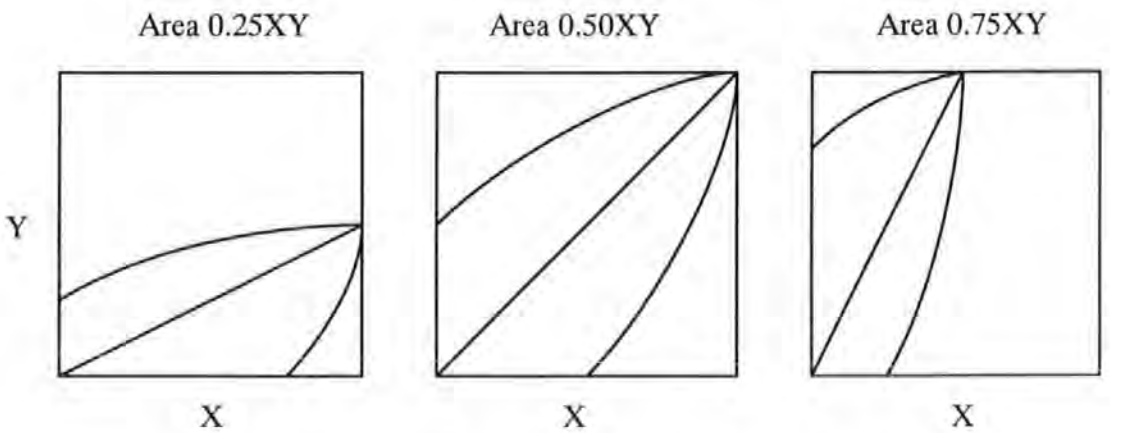


Figure 8.2 Permitted region of first point for given AUC

Note that because of the intercepts with the x and y axis there is no straight forward expression to integrate the probability over the graph. However, it is possible to define a line, that will be called a '*spine*' (because the shapes look like leaves) from the origin to the intercept of the two boundaries so that the integral can be calculated as a sum of two integrals. The first integral is from the spine to the upper boundary, first vertically, and then horizontally, the second integral is from the spine to the right boundary, first horizontally and then vertically.

Given a location x_0,y_0 for the first point, the bounds on the location of the second point x_1y_1 can be determined. This is sketched in Figure 8.3.

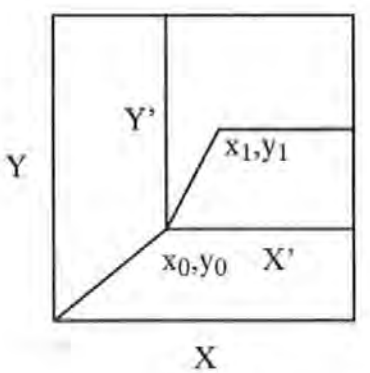


Figure 8.3 Minimum areas of second point

It can be seen that determining the location of point x_1,y_1 is recursively the same problem as determining the location of point x_0,y_0. Hence the use of the constants X and Y, instead of 1 and 1 as the width and height of the ROC graph.

The location of the last point needs a different solution. The line terminates at coordinate 1, 1 (by definition) and so the last two segments of the curve must form a right angled quadrilateral as sketched in Figure 8.4.

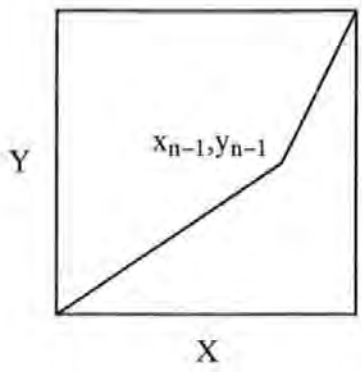


Figure 8.4 Area of last point

The area, a, of this quadrilateral is given by:

$$a = \frac{x_{n-1}\, y_{n-1}}{2} + \frac{(Y + y_{n-1})\ (X - x_{n-1})}{2}$$

Therefore, x_{n-1} is given by:

$$x_{n-1} = \frac{2a - XY - y_{n-1}X}{Y}$$

The solutions to the equation are sketched in Figure 8.5 for a = 0.25XY, 0.50XY, 0.75XY

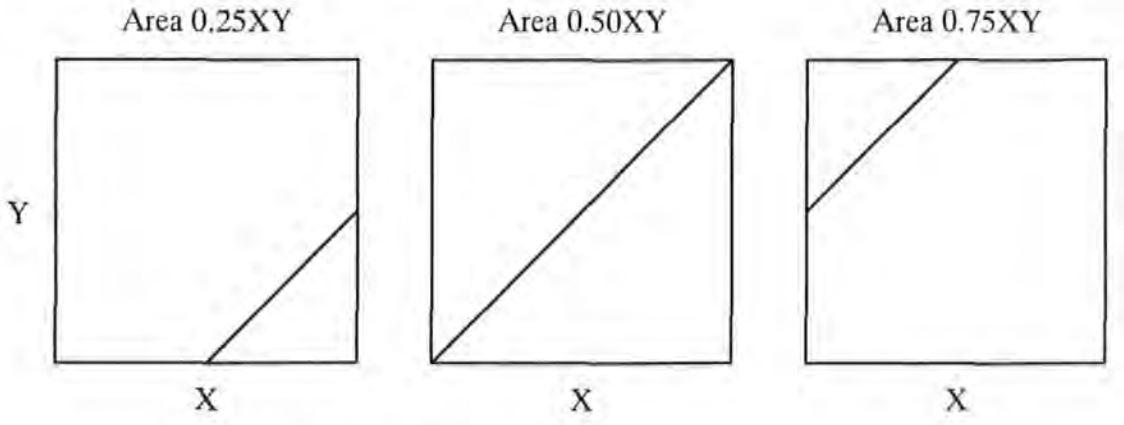


Figure 8.5 Permitted regions (lines) of last point for given areas

In order to integrate the probability along the line, it is necessary to know the bounds. If the area is less than $0.5XY$ then the bounds are $x_{n-1} = \frac{2a - XY}{Y}$ to $x_{n-1} = 1$. If the area is greater or equal to $0.5XY$ then the bounds are $x_{n-1} = 0$ to $x_{n-1} = \frac{2a - 2XY}{Y}$. The existence of such a decision at the bottom of the recursion presents a problem, but there is a solution. (Specifying the bounds in terms of y_{n-1} gives an equivalent problem.)

If the area of the last point is specified as being less than $0.5XY$ then the limits of the last integration are known. The leaf shaped bounds of the previous point are already known given the constraint that the last area is between 0 and XY. However, the last area can now be constrained to the range 0 to $0.5XY$. The $a = 0$ boundary already exists. The $a = 0.5XY$ boundary can now be derived using the same calculation as was applied to the last point. Applying the same logic, if the area of the last point is greater or equal to $0.5XY$, the boundaries have the range $a = 0.5XY$ to $a = XY$. Figure 8.6 shows this $0.5XY$ boundary for the penultimate point, when the area is $0.25XY$, $0.50XY$ and $0.75XY$. The Figure can be seen to be derived by overlaying Figure 8.2 on Figure 8.5.

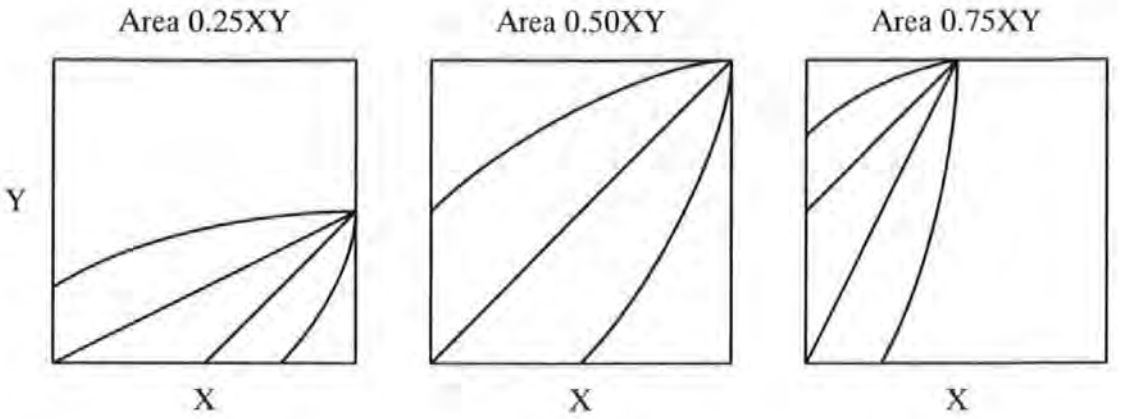


Figure 8.6 Triple regions of first point for given areas

The bookkeeping now required to integrate over the '*leaf*' becomes even more complex than it was before. However, the same logic applies to the preceding point and so on, winding back the recursion.

Thus the algorithm must divide the ROC curve into three regions at the first point, and recursively descend for each region. At the last point of every branch of the recursion it will be known if the area is less than $0.5XY$ or not. The total integration of the AUC is thus the sum of 3^n integrations.

Finally, it would be desirable to integrate the function over a small range of the AUC in order to providing a continuous pdf, rather than sampling the pdf at small intervals as implied above.

Even though this analytic solution runs in time proportional to the third power of the number of ROC points it may still be useful. The algorithmic solution runs it time proportional to the fifth power of the grid size of the quantized AUC pdf, and so there will be a balance point, under which the analytic solution is faster, and above which the algorithmic solution is faster. Thus both methods may eventually be used side by side.

8.3.3 Analytic Solution for Confusion Matrix pdf

While the algorithmic solution for calculating the pdf of a weighted confusion matrix is reasonably efficient its accuracy suffers because of the inevitable inexact quantization of the weights. It is therefore desirable to have an analytic solution to the problem.

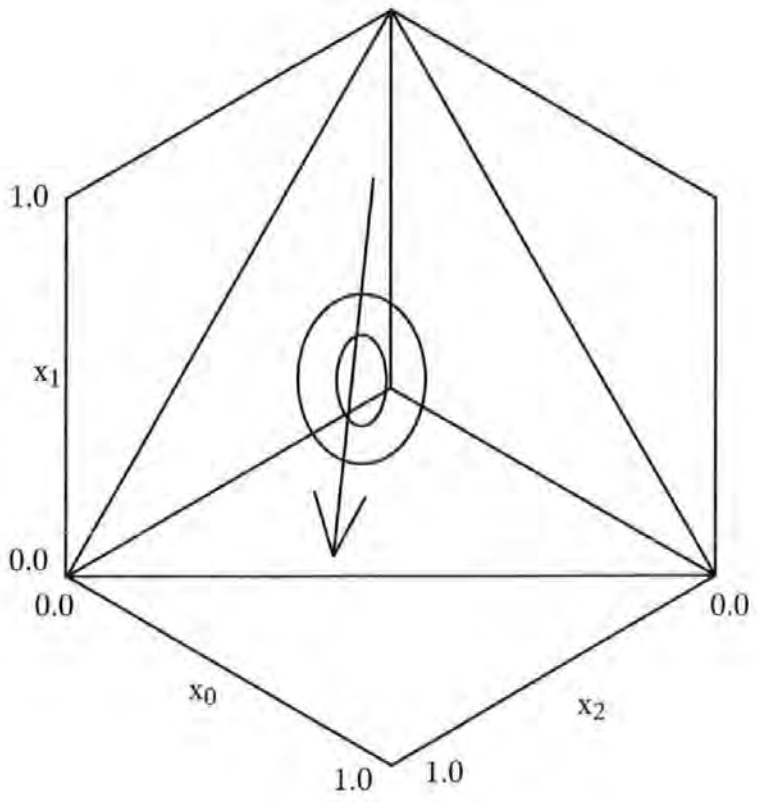


Figure 8.7 Pdf of 3 weight confusion matrix showing isoweight arrow

Figure 8.7 shows the pdf of a three weight confusion matrix, as a two dimensional (hyper)plane in a three dimensional (hyper)cube.

If the weight of each cell is given by:

$w_0 = 1.0,$ $w_1 = 0.3,$ $w_2 = 0.0$

The weight at any point is given by:

$$W = w_0x_0 + w_1x_1 + w_2x_2 = 1 \times x_0 + w_1x_1 + 0 \times x_2$$

$$\therefore W = x_0 + w_1x_1$$

The arrow in figure 8.7 shows a line of equal weight. If the pdf can be integrated over isoweight contours parallel to the arrow then the pdf of the weight of the confusion matrix will be obtained.

The diagram can be re-drawn in two dimensions, x_0 against x_1, and the diagram sheered so that isoweight contours run vertically as in Figure 8.8.

It can now be seen that the pdf of the weight between 0 and w_1 (0.3) can be obtained by integrating:

$$p_w = \int_0^{\frac{x_0}{w_1}} x_0^{a_0}\, x_1^{a_1}\, (1 - x_0 - x_1)^{a_2}\, dx$$

and from w_1 (0.3) to 1.0, by integrating:

$$p_w = \int_0^{\frac{1-x_0}{1-w_1}} x_0^{a_0}\, x_1^{a_1}\, (1 - x_0 - x_1)^{a_2}\, dx$$

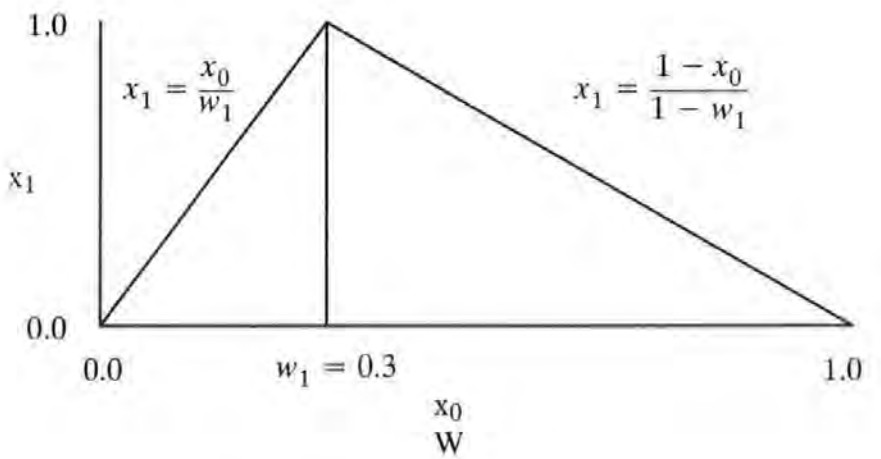


Figure 8.8 Sheered triangle for two weights

If there are four weights, the pdf is a 3 dimensional hyperplane in a 4 dimensional hypercube. The 3 dimensional hyperplane can be redrawn as a tetrahedron, and as with Figure 8.8, sheered so isoweight contours are at right angles to the x_0 axis. If w_2 is 0.8 (the last weight is now w_3 with a weight of 0.0), the tetrahedron is shown in Figure 8.9.

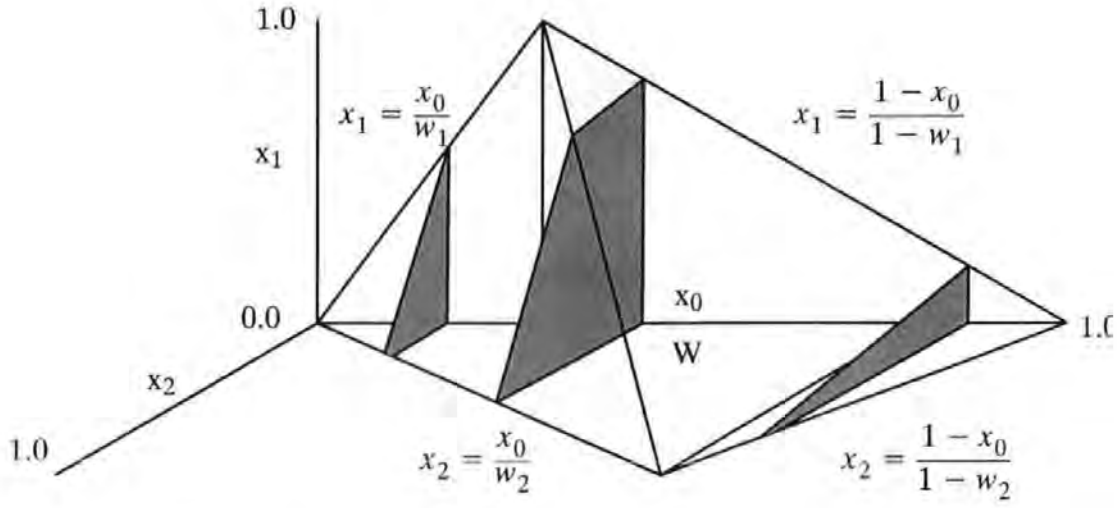


Figure 8.9 Sheered tetrahedron for three weights

There are now three regions, from $W = 0$ to $W = w_1 = 0.3$, from $W = w_1$ to $W = w_2 = 0.8$, and from $W = w_2$ to $W = 1.0$. Isoweight slices through the tetrahedron are triangular in the first and last sections, and a quadrilateral (with one right angle corner) in the mid section. The equations of the vertices of the tetrahedron on the x_1 and x_2 plane follow directly from the previous example. The bounds of the triangles and quadrilateral can thus be calculated and the sections of the pdf integrated. The mid section can be integrated by using the trapezoid rule.

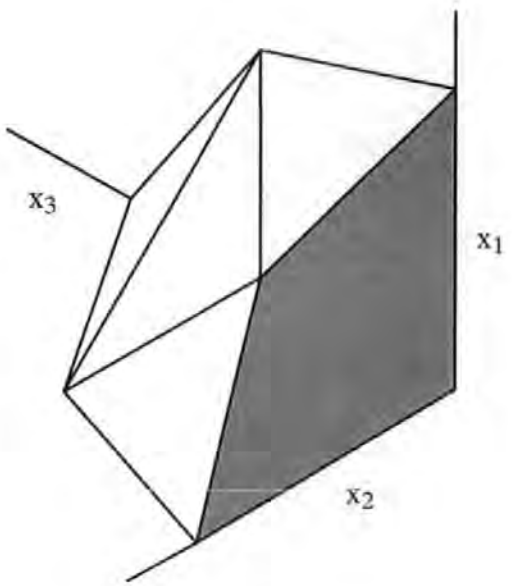


Figure 8.10 3D section through sheered hypertetrahedron

Now consider a confusion matrix with five weights. The pdf of a 5 weight confusion matrix exists on a 4 dimensional hyperplane in a 5 dimensional hypercube. It is thus only possible to illustrate this with a 3 dimensional slice through the 4 dimensional pdf, as shown in Figure 8.10. There are 4 regions, the region where the weight has the range 0 and w_1, w_1 to w_2, w_2 to w_3, and w_3 to 1.0 (the last weight is now w_4 which is 0.0).

The first and fourth regions are 4D tetrahedrons, so the integration is straight forward. Slices through the 4D second and third regions are 3D octahedrons with a right angle corner as shown in Figure 8.10. Again the vertices and surfaces can be calculated, and the integration of the pdf calculated by dividing the slices into trapeziums and summing the integral over each part. Slicing the octahedron results in three slice shapes, triangular, quadrilateral and pentagonal (each with a right angle corner), which require one, two and three further triangular/trapezoid slices.

As the number of dimensions increases there is a combinational explosion of the number of elements that have to be integrated and summed. Given the simplicity of the factorising algorithm it was not considered fruitful to pursue this approach earlier. In light of the sensi-

tivity of the factorising algorithm to quantization errors of the weights, this complex integration may prove to give a better answer in the long run.

8.3.4 Analytic Solution for Parametric pdf

It would also be very desirable to have an analytic method for generating the pdf of the AUC, and parameters, of a parametric ROC curve, since the computational method requires a large amount of computer resources. At this stage, no guidance can be offered for a solution.

8.3.5 Chance Correction

8.3.5.1 Confusion Matrices

The advantage of kappa over chi–squared is that it is chance corrected. The pdf of a confusion matrix, as calculated in section 5.6, is not chance corrected. Chance correction compares the actual probability of each cell with the joint probability of the column and row of the cell, as explained in section 4.1. The joint probability is what would be expected by chance if the test result was totally independent of the 'Gold Standard'. The simplest confusion matrix is the 2 × 2 case, as shown in Figure 8.11. The figure shows the joint probabilities of the cells z_0 to z_3 in terms of the column, x, and row, y, probabilities.

$1-y$	$z_0 = x\ (1-y)$	$z_2 = (1-x)\ (1-y)$
y	$z_3 = x\ y$	$z_1 = (1-x)\ y$
	x	$1-x$

Figure 8.11 Constraints on 2 by 2 confusion matrix

The pdf of a four cell confusion matrix is a three dimensional hyperplane in a four dimensional hypercube. The 3D hyperplane can be plotted as a tetrahedron in z_0, z_1, z_2 space as shown in Figure 8.12. However the joint probabilities are constrained to a 2D x, y surface. The joint probability equations in Figure 8.11 map this surface into z_0, z_1, z_2 space

as a twisted surface within the tetrahedron, represented in Figure 8.12 by lines of equal x value. In the general case, the column and row joint probabilities of an m by m confusion matrix is an $m + m - 1$ dimensional hyperline, within the $m^2 - 1$ hyperplane of the unconstrained pdf, within a m^2 dimensional hypercube.

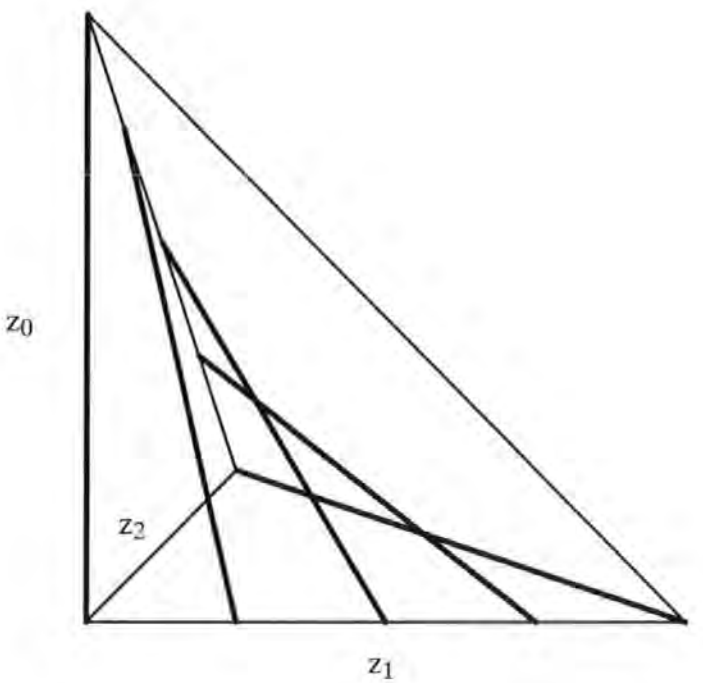


Figure 8.12 Lines of equal x value indicating joint probability surface

If the tetrahedron is integrated across slices of equal weight, the unconstrained pdf of the weights is produced. Isoweight slices will also intersect the x, y constraint plane on a line. Integrating the pdf along these lines will give the constrained pdf of the weights. The problem is deciding how to calculate the pdf of the difference between the constrained and unconstrained pdfs. It may be simply a matter of calculating the two pdfs and comparing them using the same technique as finding the difference between two ROC points (section 5.3), but this is a matter for further research.

Once a difference pdf is constructed it should then be tested using a Monte Carlo simulation. The constrained population probabilities should be calculated from the set of unconstrained population probabilities and the difference calculated.

8.3.5.2 ROC Curves

The joint probabilities of each entry in a table of ROC data can also be calculated. Figure 8.13 gives an example. There are two numbers in each entry, the first is the observed number of cases, the second, in parentheses, is the number expected by chance, calculated by multiplying the total for the row, by the total for the column, and dividing by the total of the table.

'Gold Standard'

		Diseased	Healthy	Row Sum
	Diseased	12 (6)	10 (24)	30
Test	*Unknown*	6 (4)	10 (16)	20
	Healthy	2 (8)	60 (32)	40
Column Sum		20	80	100

Figure 8.13 ROC data expected by chance

The values expected by chance can be plotted as a ROC curve. Figure 8.14 shows a plot of the data in Figure 8.13. The values expected by chance are plotted in dotted lines, while the actual values are plotted in bold lines.

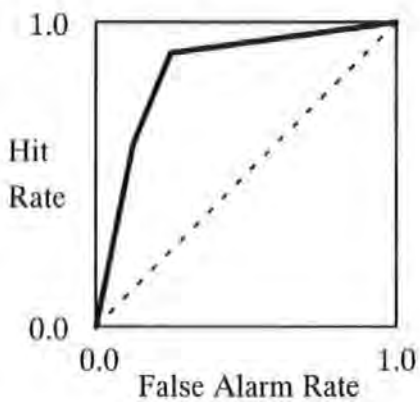


Figure 8.14 Chance corrected ROC curve

The chance line always runs across the diagonal, and corresponds to the ROC curve obtained when the healthy and diseased populations are identical, as explained in section 1.1, Figure 1.3 (b). The chance line is often illustrated (e.g. [51]) on ROC curves to show the separation of the ROC curve, and any confidence/posterior boundaries, from the

chance line. If the chance line is outside the confidence/posterior bound of a ROC curve, or point, the implication is that the detection ability of the system under test is better than chance.

However, as discussed above, a pdf can be calculated for the chance line from the sample. The pdf lies on a twisted surface within the pdf of the ROC curve. Posterior limits can therefore be calculated for the chance line itself. The practice of drawing a straight diagonal chance line on a ROC curve and use it make inferences about the detection ability of the system under test is therefore questionable. It is for this reason that the chance correction line does not appear in any of the ROC curves in Chapter 7.

It is therefore suggested that any future investigation of chance correction in confusion matrices should include chance correction in ROC curves.

8.3.6 Comparing Systems

Section 5.3 presented a method for comparing the pdfs of pairs of ROC points to generate the pdf of the difference, which can be used to objectively compare the performance of two systems.

While no other pdfs have been formally compared, it is likely that exactly the same method will work for the nonparametric and parametric AUC and for the weight of a weighted confusion matrix. However, the parameters of a ROC curve cause a tricky problem in presentation. The difference in the mean has a range of plus to minus infinity. This is mapped into a finite range by a sigmoid function to enable it to be plotted. At this stage it is not clear how to present the difference between two sigmoid plots in a way that retains any intuitive meaning. This is a matter for future research.

8.3.7 Correlated ROC Curves

Suppose there are two different tests for one disease, for instance, two distinct biochemical blood markers. The situation could equally apply to an intelligent medical system and an expert examining mammograms, but it is easier to think in terms of biochemical markers for some aspects of this discussion.

Figure 8.15 shows a plot of the blood concentration of two hypothetical biochemical markers for 20 diseased cases and 14 healthy cases. The cases have been plotted on a continuous scale, though five categories have also been assigned. The underlying probability distributions of the diseased and health populations are two dimensional Gaussian distributions as indicated by the 95% contours shown in the figure.

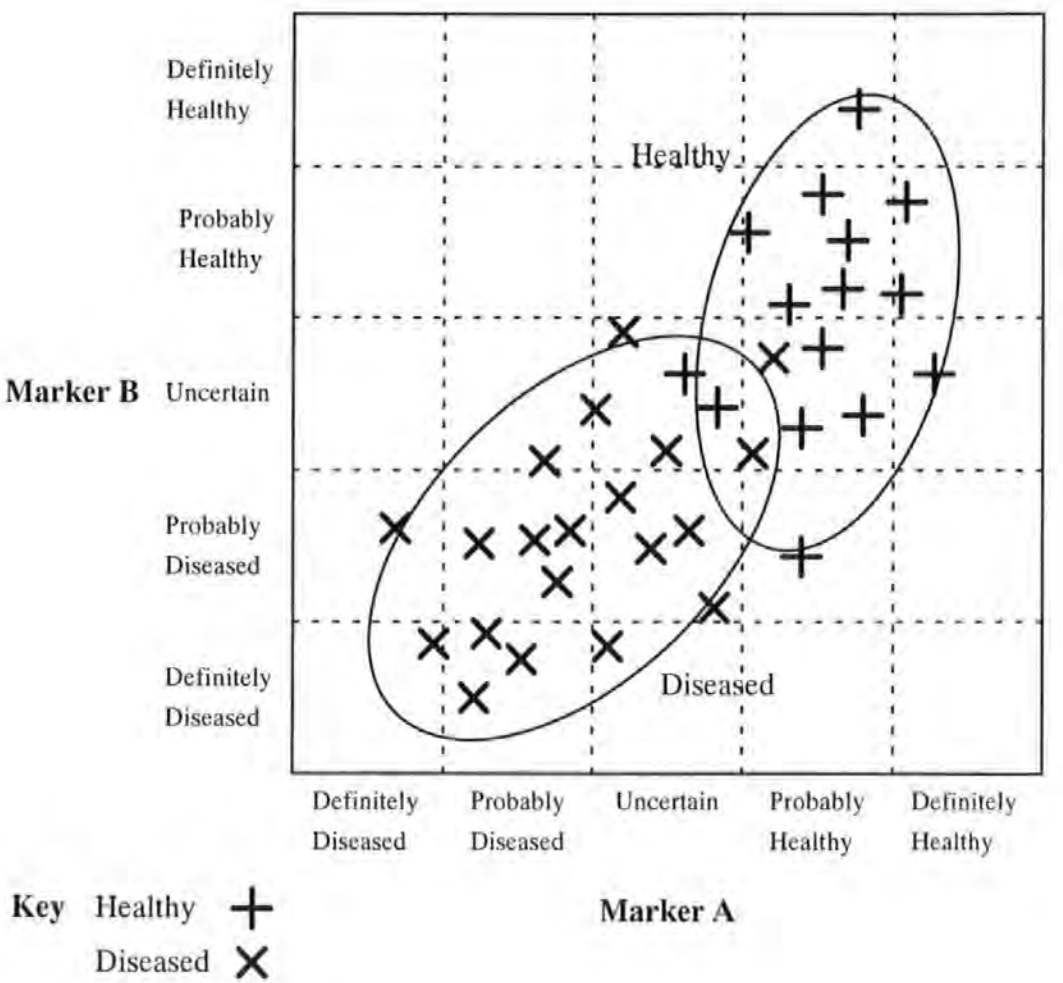


Figure 8.15 Underlying model of correlated ROC curves

Looked at from the view point of testing Marker B, the diseased and healthy cases have non-overlapping Gaussian distributions and a ROC curve can be plotted from the data. The situation is similar when looking from the point of view of testing Marker A. The two ROC curves could be compared by using the method in section 5.3 to determine which was the best marker for the disease. This treats the tests as if they had been carried out

on completely different cases. It ignores the fact that the same cases are being tested and therefore that some potentially valuable information is being lost.

The first observation is that it is possible to construct a better diagnosis by combining the results of both tests. Kramar et al. [92], produced software for doing this. It generated the linear equation of test marker results that gave optimum diagnostic discrimination.

Secondly, if markers are highly correlated, as shown in the sketch in Figure 8.16, the correlation reduces the impact of case sampling variation and hence increases the accuracy with which differences between tests can be measured. In Figure 8.16, the case shown by the black dot represents a measurement error which can be corrected for by the high correlation shown by the rest of the data set. It is therefore important to account for correlation effects in future work on measuring the differences between ROC curves.

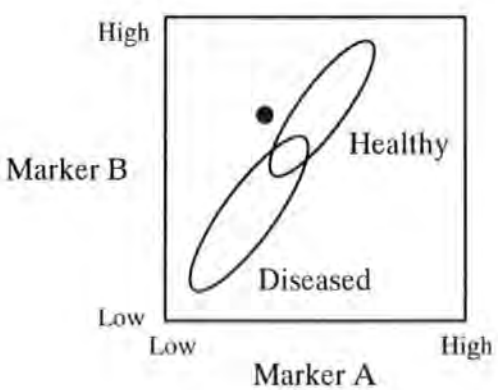


Figure 8.16 Model of highly correlated ROC curves

Maximum likelihood analysis methods can be used to estimate the parameters of both ROC curves and the correlation coefficient. Fully paired tests are used when every case has been evaluated with both tests [93]. However, sometimes this is not possible and some cases can only be evaluated with one test. Rather than having to discard such cases, and loose information, partially paired ROC analysis can be used [67].

Applying the novel methods introduced in Chapter 5 to all these areas may be beneficial, especially at low sample sizes.

8.3.8 Sources of Error and Bias

There are many errors and bias that occur in analysing real systems, for instance variation in equipment, variation between technicians, and biologic variability. Correcting for these errors is a current area of research.

Verification bias occurs when the selection of cases for the ROC analysis of a diagnostic test is influenced by the results of the test itself. Rodenberg and Zhou [94], report a study in which 4706 people in the two age groups, 65 to 75, and 75+, were screened for dementia by the test under investigation. A random sample of these cases was then examined in detail by dementia experts to determine the 'Gold Standard' for these cases. However, for ethical reasons, the random sample was biased towards cases that had already tested positive for dementia. Rodenberg and Zhou present statistical methods for correcting for this verification bias.

Chan et al. [95] observed that if an intelligent medical system is constructed by machine learning techniques from a sample of cases drawn from the population, the sampling of those cases will exhibit exactly the same statistical variation as another sample of cases drawn from the same population used to test the system. There is thus a double error compared with an 'ideal' system produced and tested with an infinite sample.

Kim and Gleser [96] have developed methods for estimating the true AUC of both parametric and nonparametric tests, when the test measurements have errors of known variance. The measurement error can be estimated by taking duplicate measurements of the same subject. Schisterman et al. [97], provide another method for parametric ROC curves.

It would be desirable to investigate whether the new methods presented in this thesis can give any insight to these problems.

8.3.9 Sample Sizes Calculations

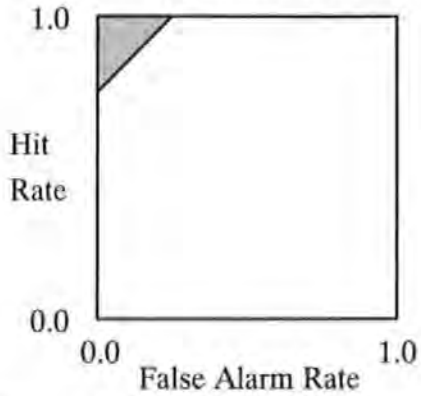


Figure 8.17 System performance requirement

Suppose that a system is required where one population ROC point is inside the shaded area at the top left of the ROC graph of Figure 8.17. How many cases have to be collected to prove (or disprove) this with 95% confidence?

Tables of sample sizes based on existing methods are available [98]. If the novel methods presented in this thesis are to reach their potential, methods for generating this information will have to be investigated.

8.3.10 Collecting Data

While the research presented so far in this thesis has focused sharply on one tiny aspect of the vast field of intelligent medical system evaluation, consideration needs to be given to how this work can best be applied.

Appendix D presents some ideas on how an intelligent medical system can be divided into modules to make it easier for data to be collected. Each module can then be objectively compared with experts, using the statistical methods presented here. Further work on this could be done as a separate project.

8.4 Conclusions

This thesis has presented new methods for calculating the exact posterior intervals for ROC curves and confusion matrices for any sample size. These are vital for proper evaluation of intelligent medical systems where the number of test cases is severely restricted and are too low to obtain accurate results using existing statistics. The techniques used may also provide a paradigm for other statistical problems.

References

1 P. Sandiford, H. Annett, R. Cibulskis, 'What can information systems do for primary health care? An international perspective', Social Science and Medicine, Vol.34, No.10, pp.1077–1087, 1992.

2 H. W. Gottinger, 'Technology assessment and forecasting of medical expert systems (MEST)', Methods of Information in Medicine., Vol.27, pp.58–66, 1988.

3 A. K. Szczepura, J. A. Stilwell, 'Information for decision makers at hospital laboratory level: an example of the graphical method of representing costs and effects for replacement automated technology in a haematology laboratory', Social Science and Medicine, Vol.26, No.7, pp.715–725, 1988.

4 C. J. Green, M. M. Keyes, 'Verification and validation of expert systems', in: Proc: Western Conf. on expert systems, Anaheim (IEEE Computer Society), pp.38–43, 1987.

5 R. M. O'Keefe, O. Baici, E. P. Smith, 'Validating Expert System Performance', IEEE Expert, Winter, pp.81–90, 1987.

6 P. L. Miller, D. F. Sittig, 'The evaluation of clinical decision support systems: what is necessary verses what is interesting', Medical Informatics, Vol.15, No.3, pp.185–190, 1990.

7 R. S. Sharma, D. W. Conrath, 'Some soft measures for performance analysis: The 'core' dimensions of expert system quality', Microelectron. Reliab. Vol.36, No.6, pp.775–796, 1993.

8 L. Kingsland, G. Sharp, R. Capps, J. Benge, D. Kay, G. Reese, S. Hazelwood, D. Lindberg, 'Testing a criteria–based consulting system in rheumatology', Proceedings of MedInfo–83, pp.514–517, 1983.

9 H. J. Moens, 'Validation of the AI/RHEUM Knowledge Base with Data from Consecutive Rheumatological Outpatients', Methods of Information in Medicine, Vol.31, pp.175–181, 1992.

10 K. P. Adlassnig, W. Scheithauer, 'Performance evaluation of medical expert systems using ROC curves', Computers and Biomedical Research, Vol.22, No.4, pp.297–313, 1989.

11 J. S. Kippenham, W. W. Barker, S. Pascal, J. Nagel, R, Duara, 'Evaluation of a Neural–Network Classifier for PET Scans of Normal and Alzheimer's Disease Subjects', The Journal of Nuclear Medicine, Vol.33, No.8, pp.1459–1467, 1992.

12 M. Kuhn, T. Zemmler, M. Reichert, D. Rosner, O. Baumiller, H. Knapp, 'An Integrated Knowledge–Based System to Guide the Physician During Structured Reporting', Methods of Information in Medicine, Vol.33, pp.417–422, 1994.

13 J. Cohen, 'A coefficient of agreement for nominal scales', Educational and Psychological Measurement, Vol.20, No.1, pp.37–46, 1960.

14 J. Cohen, 'Weighted Kappa: nominal scale agreement with provision for scaled disagreement or partial credit', Psychological Bulletin, Vol.70, No.4. pp.213–220, 1968.

15 Gelman, A., Carlin, J. B., Stern, H. S., Rubin, D. B., 'Bayesian Data Analysis', Chapman and Hall, London, ISBN 0 412 03991 5, 1995.

16 A. R. van Erkel, P. M. T. Pattynama, 'Receiver operating characteristic (ROC) analysis: Basic principles and applications in radiology', European Journal of Radiology, Vol.27, pp.88–94, 1998.

17 R. M. Centor, J. S. Schwartz, 'An Evaluation of Methods for Estimating the Area Under the Receiver Operating Characteristic (ROC) Curve', Med. Decis. Making, Vol.5, No.2, pp.149–156, 1985.

18 K. Jensen, H–H. Muller, H, Schafer, 'Regional confidence bands for ROC Curves', Statistics in Medicine, Vol.19, pp.493–509, 2000.

19 N. A. Obuchowski, M. L. Lieber, 'Confidence Intervals for the Receiver Operating Characteristic Area in Studies with Small Samples', Academic Radiology, Vol.5, pp.561–571, 1998.

20 M. Altaye, A Donner, M. Eliasziw, 'A general goodness-of-fit approach for inference procedures concerning the kappa statistic', Statistics in Medicine, Vol.20, pp.2479-2499, 2001.

21 J. A. Swets, 'Measuring the Accuracy of Diagnostic Systems', Science, Vol.240, pp.1285-1293, 1988.

22 B. G. Buchanan, E. H. Shortliffe, (eds.), 'Rule-Based Expert Systems', Addison-Wesley, Reading, MA, ISBN 0-201-10172-6,1984.

23 Y. Z. Wu, K. N. Doi, M. L. Giger, R. M. Nishikawa, 'Computerized detection of clustered microcalcifications in digital mammograms: Application of artificial neural networks', Medical Physics, Vol.19, No.3, pp.555-560, 1992.

24 M. E. Boon, E. D. Kleinschmidt-Guy, E. Ouwerkerk-Noordam, 'PAPNET for analysis of proliferating (MIB-1 positive) cell populations in cervical smears', European Journal of Morphology, Vol.32, No.1, pp.78-85, 1994.

25 S. Uckun, 'Intelligent systems in patient monitoring and therapy, a survey of research projects', International Journal of Clinical Monitoring and Computing, Vol.11, pp.241-253, 1994.

26 J. Wyatt, D. Spiegelhalter, 'Evaluating medical expert systems: what to test and how?', Medical Informatics, Vol.15, No.3, pp.205-217, 1990.

27 K. P. Adlassnig, G. Kolarz, W. Scheithauer, 'Present state of the medical expert system CADIAG 2', Methods of Information in Medicine, Vol.24, pp.13-20, 1985.

28 C. Whitbeck, R. Brooks, 'Criteria for evaluating a computer aid to clinical reasoning', The Journal of Medicine and Philosophy. Vol.8, pp.51-65, 1983.

29 D. S. Murphy, S. A. Yetmar, 'Auditor evidence evaluation: expert systems as credible sources', Behaviour and Information Technology, Vol.15, No.1, pp.14-23, 1996.

30 L. Timnick, 'Electronic bullies', Psychology Today, Vol.16, pp.10-15, 1982.

31 A. Maynard, 'The logic of economic choice in health care', in C. I. Phillips, 'Logic in medicine', 2nd edition, BMJ publishing, London, ISBN 0-7279-0854-5,1995.

32 B. Jennett, 'High Technology Medicine and Quality of Life', International Journal of Technology Assessment in Health Care, Vol.3, No.1, pp.51-60, 1987.

33 R. Howard, 'Microrisks for medical decision analysis', International Journal of Technical Assessment in Health Care, Vol.5, pp.357-370, 1989.

34 K. Clarke, R. O'Moore, R. Smeets, J. Talmon, J. Brender, P. McNair, P. Nykanen, J. Grimson, B. Barber, 'A methodology for evaluation of knowledge-based systems in medicine', Artificial Intelligence in Medicine, Vol.6, pp.107-121, 1994.

35 C. E. Phelps, A. Hutson, 'Estimating Diagnostic Test Accuracy Using a 'Fuzzy Gold Standard'', Medical Decision Making, Vol.15, No.1, pp.44-57, 1995.

36 E. T. Whittaker, G. N. Watson, 'A Course of Modern Analysis', 4th edition, Cambridge University Press, Cambridge, p.258, 1927.

37 R. A. Fisher, [No title available], Proc. Camb. Phil. Soc. 26, p.528, 1930, cited in [40].

38 R. A. Fisher, [No title available], Proc. Roy. Soc. A 139, p.343, 1933, cited in [40].

39 J. Neyman, 'On the two different aspects of the representative method: the method of stratified sampling and the method of purposive selection', Journal of Royal Statistical Society, Vol.XCVIII, pp.558-606, 1934.

40 C. J. Clopper, E. S. Pearson, 'The Use of Confidence or Fiducial Limits Illustrated in the Case of the Binomial', Biometrika, Vol.26, pp.404-413, 1934.

41 R. B. Murphy, 'Non-parametric Tolerance Limits', Ann. Math. Statistics, Vol.19, pp.581-589, 1948.

42 R. A. Hilgers, 'Distribution-free confidence bounds for ROC curves', Methods of Information in Medicine, Vol.30, pp.137-150, 1991.

43 Barnett, V., 'Comparative Statistical Inference', Wiley and Sons, London, ISBN 0 471 05401 1, 1975.

44 K. H. Zou, W. J. Hall, D. E. Shapiro, 'Smooth non-parametric receiver operating characteristic (ROC) curves for continuous diagnostic tests', Statistics in Medicine, Vol.16, pp.2143-2156, 1997.

45 J. B. Tilbury, P. W. J. Van Eetvelt, J. M. Garibaldi, J. S. H. Curnow, E. C. Ifeachor, 'Receiver Operating Characteristic Analysis for Intelligent Medical Systems - A New Approach for Finding Confidence Intervals', IEEE Transactions on Biomedical Engineering, Vol. 47, No.7, pp.952-963, 2000.

46 F. Wilcoxon, 'Individual comparisons by ranking methods', Biometrics, Vol.1, pp.80-83, 1945.

47 H. B. Mann, D. R. Whitney, 'On a test of whether one of two random variables is stochastically larger than the other', Annals of Mathematical Statistics, Vol.18, pp.50-60, 1947.

48 D. Bamber, 'The area above the ordinal dominance graph and the area below the receiver operating graph', J. Math. Psychol., Vol.12, pp.387-415, 1975.

49 D. Stirzaker, 'Probability and Random Variables - a beginner's guides', Cambridge University Press, Cambridge, pp.164-165, 1999.

50 A. de Moivre, 'Approximatio ad summam terminorum binomii $(a+b)^n$ in seriem expansi', suppliment to 'Miscellanea Analytica', London, 1756.

51 J. A. Hanley, B. J. McNeil, 'The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve', Diagnostic Radiology, Vol.143, No.1, pp.29-36, 1982.

52 E. R. DeLong, D. M. DeLong, D. L. Clarke-Pearson, 'Comparing the Areas Under Two or More Correlated Receiver Operating Characteristic Curves: A Nonparametric Approach', Biometrics, Vol.44, pp.837-845, 1988.

53 J. A. Hanley, K. O. Hajian-Tilaki, 'Sampling Variability of Nonparametric Estimates of the Areas under Receiver Operating Characteristic Curves: An Update', Academic Radiology, Vol.4, pp.49-58, 1997.

54 B. J. McNeil, J. A. Hanley, 'Statistical approaches to the analysis of receiver operating characteristic (ROC) curves', Medical Decision Making, Vol.4, pp.137-150, 1984.

55 D. D. Dorfman, K. S. Berbaum, C. E. Metz, 'Receiver operating characteristic rating analysis: generalization to the population of readers and patients with the jackknife method', Invest. Radiol., Vol.27, pp.723-731, 1992.

56 W. Hoeffding, 'A class of statistics with asymptotically normal distributions', The Annals of Mathematical Statistics, Vol.19, pp.293-325, 1948.

57 C. E. Metz, 'Basic Principles of ROC Analysis', Seminars in Nuclear Medicine, Vol.8, pp.283-298, 1978.

58 D. D. Dorfman, E. Alf, 'Maximum-Likelihood Estimation of Parameters of Signal-Detection Theory - a direct solution', Psychometrika, Vol.33, pp.117-124, 1968.

59 M. G. Kendell, A. Stuart, J. K. Ord, 'The advanced theory of statistics', Griffin, London, Vol.1, p.284, 5th Edition, 1987.

60 D. D. Dorfman, E. Alf, 'Maximum-Likelihood Estimation of Parameters of Signal-Detection Theory and Determination of Confidence Intervals - Rating-Method Data', Journal of Mathematical Psychology, Vol.6, pp.487-496, 1969.

61 C. E. Metz, B. A. Herman, J. Shen, 'Maximum Likelihood estimation of receiver operating characteristic (ROC) curves from continuously-distributed data', Statistics in Medicine, Vol.17, pp.1033-1053, 1998.

62 C. R. Rao, 'Advanced statistical methods in biometric research', Wiley, New York, 1952.

63 X. Pan, C. E. Metz, 'The 'Proper' Binormal Model: Parametric Receiver Operating Characteristic Curve Estimation with Degenerate Data', Academic Radiology, Vol.4, No.5, pp.380–389, 1997.

64 C. E. Metz, H. B. Kronman, 'Statistical significance tests for binormal ROC curves', Journal of Mathematical Psychology, Vol.22, pp.218–243, 1980.

65 D. D. Dorfman, K. S. Berbaum, C. E. Metx, R. V. Lenth, J. A. Hanley, H. A. Dagga, 'Proper Receiver Operating Characteristic Analysis: The Bigamma Model', Statistics in Radiology, Vol.4, pp.138–149, 1996.

66 K. O. Hijian–Tilaki, J. A. Hanley, L. Joseph, J.Collet, 'A Comparison of Parametric and Nonparametric Approaches to ROC Analysis of Quantitative Diagnostic Tests, Medical Decision making, Vol.17, pp.94–102, 1997.

67 C. E. Metz, B. A. Herman, C. A. Roe, 'Statistical Comparison of Two ROC–curve Estimates Obtained from Partially–paired Datasets', Medical Decision Making, Vol.18, No.1, 1998.

68 N. A. Obuchowski, 'Nonparametric Analysis of Clustered ROC Curve Data', Biometrics, Vol.53, pp.567–578, 1997.

69 N. A. Obuchowski, D. K, McClish, 'Sample Size Determination for Diagnostic Accuracy Studies Involving Binormal ROC Curve Indices', Statistics in medicine, Vol.16, pp.1529–1542, 1997.

70 B. Reiser, D. Faraggi, 'Confidence Intervals for the Generalized ROC Criterion', Biometrics, Vol.53, pp.644–652, 1997.

71 C. A. Roe, C. E. Metz, 'Dorfman–Berbaum–Metz Method for Statistical Analysis of Multireader, Multimodality Receiver Operating Characteristic Data: Validation with Computer Simulation', Statistics in Radiology, Vol.4, No.4, pp.298–303, 1997.

72 H. Schafer, 'Efficient confidence bounds for ROC curves', Statistics in Medicine, Vol.13, pp.1551–1561, 1994.

73 H. H. Song, 'Analysis of Correlated ROC Areas in Diagnostic Testing', Biometrics Vol.53, pp.370–382, 1997.

74 M. Sullivan Pepe, 'Three Approaches to Regression Analysis of Receiver Operating Characteristic Curves for Continuous Test Results', Biometrics, Vol.54, pp.124–135, 1998.

75 S. J. Walsh, 'Limitations to the robustness of binormal ROC curves: effects of model misspecification and location of decision thresholds on bias, precision, size and power', Statistics in Medicine, Vol.16, pp.669–679, 1997.

76 R. J. Landis, G. G. Koch, 'The Measurement of Observer Agreement for Categorical Data', Biometrics, Vol.33, pp.159–174, 1977.

77 N. J.–M. Blackman, J. J. Koval, 'Interval estimation for Cohen's kappa as a measure of agreement', Statistics in Medicine, Vol.19, pp.723–741, 2000.

78 SAS Institute Inc, 'SAS/STAT Software: Changes and Enhancements through Release 6.23', SAS Institute Inc, Cary NC, 1997.

79 J. L. Fleiss, J. Cohen, B. S. Everitt, 'Large–sample standard errors of kappa and weighted kappa', Psychological Bulletin, Vol.72, pp.323–327, 1969.

80 J. B. Garner, 'The standard error of Cohen's kappa', Statistics in Medicine, Vol.10, pp.767–775, 1991.

81 M. H. Quenouille, 'Notes on bias estimation', Biometika, Vol.43, pp.353–360, 1956.

82 J. W. Tukey, 'Bias and confidence intervals in not quite large samples', Annals of Mathematical Statistics, Vol.29, pp.614–616, 1958.

83 R. G. Miller, 'A trustworthy jack–knife', Annals of Mathematical Statistics, Vol.35, pp.1594–1605, 1964.

84 M. Abramowitz, I. A. Stegum, (eds.), 'Handbook of Mathematical Functions, with Formulas, Graphs, and Mathematical Tables', Dover, New York, p.258, 1968.

85 S. L. Moshier, 'Methods and programs for mathematical functions', Ellis Horwood, Chichester, 1989.

86 R. F. Wagner, S. V. Beiden, C. E. Metz, 'Continuous versus Categorical Data for ROC Analysis: Some Quantitative Considerations', Academic Radiology, Vol.8, No.4, 2001.

87 R. D. Baker, J. B. Tilbury, 'Rapid computation of the permutation paired and grouped T-tests', Applied Statistics, Vol.42, No.2, pp.431-441, 1993.

88 P. L'Ecuyer, 'Efficient and Portable Combined Random Number Generators', Communications of the ACM, Vol.31, No.6, pp.742-774, 1988.

89 D. M. Green, J. Swets, 'Signal detection theory and psychophysics', Wiley, New York, pp.45-49, 1966.

90 P. A. Herzberg, 'Principles of Statistics', Wiley, New York, 1983, p.279.

91 N. R. Neave, 'Statistics Tables for mathematicians, engineers, economists and the behavioural and management sciences', Unwin Hyman, London, 1978.

92 A. Kramar, D. Faraggi, A, Fortune, B. Reiser, 'mROC: a computer program for combining tumour markers in predicting disease states', Computer Methods and Programs in Biomedicine, Vol.66, pp.199-207, 2001.

93 J. A. Hanley, B. J. McNeil, 'A method of comparing the areas under receiver operating characteristic curves derived from the same cases', Radiology, Vol.148, p.839, 1983.

94 C. Rodenberg, X. Zhou, 'ROC Curve Estimation When Covariates Affect the Verification Process', Biometrics, Vol.56, pp.1256-1262, 2000.

95 H. Chan, R. F. Wagner, N. Petrick, 'Classifier design for computer-aided diagnosis: Effect of finite sample size on the mean performance of classical and neural network classifiers', Medical Physics, Vol.26, No.12, 1999.

96 J. Kim, L. J. Gleser, 'Simex approaches to measurement error in ROC studies', Communications in Statistics - Theory and Methods, Vol.29, No.11, pp.2473-2491, 2000.

97 E. F. Schisterman, D. Faraggi, B. Reiser, M. Trevisan, 'Statistical Inference for the Area under the Receiver Operating Characteristic Curve in the Presence of Random Measurement Error, American Journal of Epidemiology, Vol.154, No.2, pp.174-179, 2001.

98 N. A. Obuchowski, 'Sample Size Tables For Receiver Operating Characteristic Studies', American Journal of Rontgenology, Vol.175, No.3, pp.603-608, 2000.

A Mathematical Derivations

A.1 Changing the Beta Function Limits

In general, the limits required of the Beta function are 0 to u, not 0 to 1, so let:

$p = u \cdot t$ which implies $dp = u \cdot dt$

$$\therefore \int_0^u p^M \cdot (u - p)^N \, dp = \int_0^1 (u \cdot t)^M \cdot (u - u \cdot t)^N \cdot u \, dt$$

$$\therefore \int_0^u p^M \cdot (u - p)^N \, dp = \int_0^1 u^M \cdot t^M \cdot (u \cdot (1 - t))^N \cdot u \, dt$$

$$\therefore \int_0^u p^M \cdot (u - p)^N \, dp = \int_0^1 u^M \cdot t^M \cdot u^N \cdot (1 - t)^N \cdot u \, dt$$

$$\therefore \int_0^u p^M \cdot (u - p)^N \, dp = \int_0^1 u^M \cdot u^N \cdot u \cdot t^M \cdot (1 - t)^N \, dt$$

$$\therefore \int_0^u p^M \cdot (u - p)^N \, dp = u^M \cdot u^N \cdot u \cdot \int_0^1 t^M \cdot (1 - t)^N \, dt$$

$$\therefore \int_0^u p^M \cdot (u - p)^N \, dp = u^{M+N+1} \cdot \int_0^1 t^M \cdot (1 - t)^N \, dt$$

$$\therefore \int_0^u p^M \cdot (u - p)^N \, dp = u^{(M+N+1)} \cdot \frac{M! \; N!}{(M + N + 1)!}$$

A.2 Integrals for One to Four ROC Points

A.2.1 One ROC Point

$f(s) = s^{a_0} \; (1 - s)^{a_1}$

A.2.2 Two ROC Points

$$f(s) = \int_0^{1-s} s^{a_0}\, x_1^{a_1}\, (1 - s - x_1)^{a_2}\, dx_1$$

$$f(s) = \int_0^{s} x_0^{a_0}\, (s - x_0)^{a_1}\, (1 - s)^{a_2}\, dx_0$$

A.2.3 Three ROC Points

$$f(s) = \int_0^{1-s} \int_0^{1-s-x_1} s^{a_0}\, x_1^{a_1}\, x_2^{a_2}\, (1 - s - x_1 - x_2)^{a_3}\, dx_2\, dx_1$$

$$f(s) = \int_0^{s} \int_0^{1-s} x_0^{a_0}\, (s - x_0)^{a_1}\, x_2^{a_2}\, (1 - s - x_2)^{a_3}\, dx_2\, dx_0$$

$$f(s) = \int_0^{s} \int_0^{s-x_0} x_0^{a_0}\, x_1^{a_1}\, (s - x_0 - x_1)^{a_2}\, (1 - s)^{a_3}\, dx_1\, dx_0$$

A.2.4 Four ROC Points

$$f(s) = \int_0^{1-s} \int_0^{1-s-x_1} \int_0^{1-s-x_1-x_2} s^{a_0}\, x_1^{a_1}\, x_2^{a_2}\, x_3^{a_3}\, (1 - s - x_1 - x_2 - x_3)^{a_4} dx_3 dx_2 dx_1$$

$$f(s) = \int_0^{s} \int_0^{1-s} \int_0^{1-s-x_2} x_0^{a_0}\, (s - x_0)^{a_1}\, x_2^{a_2}\, x_3^{a_3}\, (1 - s - x_2 - x_3)^{a_4}\, dx_3\, dx_2\, dx_0$$

$$f(s) = \int_0^{s} \int_0^{s-x_0} \int_0^{1-s} x_0^{a_0}\, x_1^{a_1}\, (s - x_0 - x_1)^{a_2}\, x_3^{a_3}\, (1 - s - x_3)^{a_4}\, dx_3\, dx_1\, dx_0$$

$$f(s) = \int_0^{s} \int_0^{s-x_0} \int_0^{s-x_0-x_1} x_0^{a_0}\, x_1^{a_1}\, x_2^{a_2}\, (s - x_0 - x_1 - x_2)^{a_3}\, (1 - s)^{a_4}\, dx_2\, dx_1\, dx_0$$

A.3 Solutions for Four Point ROC Curve

A.3.1 Four ROC Points, 1st Point

$$p(s_0) = \int_0^{1-s_0} \int_0^{1-s_0-x_1} \int_0^{1-s_0-x_1-x_2} {s_0}^{a_0}\ {x_1}^{a_1}\ {x_2}^{a_2}\ {x_3}^{a_3}\ (1 - s - x_1 - x_2 - x_3)^{a_4}\ dx_3$$

Applying the Beta function with limits 0 to s, where s is in turn $1 - s_0 - x_1 - x_2$, $1 - s_0 - x_1$ and $1 - s_0$:

$$f(s_0) = \int_0^{1-s_0} \int_0^{1-s_0-x_1} {s_0}^{a_0}\ {x_1}^{a_1}\ {x_2}^{a_2}\ (1 - s_0 - x_1 - x_2)^{a_3+a_4+1}\ \frac{a_3!\ a_4!}{(a_3 + a_4 + 1)!}\ dx_2 dx_1$$

$$f(s_0) = \int_0^{1-s_0} {s_0}^{a_0}\ {x_1}^{a_1}\ (1 - s_0 - x_1)^{a_2+a_3+a_4+2}\ \frac{a_2!\ (a_3 + a_4 + 1)!}{(a_2 + a_3 + a_4 + 2)!}\ \frac{a_3!\ a_4!}{(a_3 + a_4 + 1)!}\ dx_1$$

$$f(s_0) = {s_0}^{a_0}\ (1 - s_0)^{a_1+a_2+a_3+a_4+3}\ \frac{a_1!\ (a_2 + a_3 + a_4 + 2)!}{(a_1 + a_2 + a_3 + a_4 + 3)!}\ \frac{a_2!\ a_3!\ a_4!}{(a_2 + a_3 + a_4 + 2)!}$$

$$f(s_0) = {s_0}^{a_0}\ (1 - s_0)^{a_1+a_2+a_3+a_4+3}\ \frac{a_1!\ a_2!\ a_3!\ a_4!}{(a_1 + a_2 + a_3 + a_4 + 3)!}$$

A.3.2 Four ROC Points, 2nd Point

$$f(s_1) = \int_0^{s_1} \int_0^{1-s_1} \int_0^{1-s_1-x_2} {x_0}^{a_0}\ (s_1 - x_0)^{a_1}\ {x_2}^{a_2}\ {x_3}^{a_3}\ (1 - s_1 - x_2 - x_3)^{a_4}\ dx_3 dx_2 dx_0$$

$$f(s_1) = \int_0^{s_1} \int_0^{1-s_1} {x_0}^{a_0}\ (s_1 - x_0)^{a_1}\ {x_2}^{a_2}\ (1 - s_1 - x_2)^{a_3+a_4+1}\ \frac{a_3!\ a_4!}{(a_3 + a_4 + 1)!}\ dx_2 dx_0$$

$$f(s_1) = \int_0^{s_1} {x_0}^{a_0}\ (s_1 - x_0)^{a_1}\ (1 - s_1)^{a_2+a_3+a_4+2}\ \frac{a_2!\ (a_3 + a_4 + 1)!}{(a_2 + a_3 + a_4 + 2)!}\ \frac{a_3!\ a_4!}{(a_3 + a_4 + 1)!}\ dx_0$$

$$f(s_1) = {s_1}^{a_0+a_1+1} \cdot \frac{a_0!\ a_1!}{(a_0 + a_1 + 1)!}\ (1 - s_1)^{a_2+a_3+a_4+2}\ \frac{a_2!\ a_3!\ a_4!}{(a_2 + a_3 + a_4 + 2)!}$$

A.3.3 Four ROC Points, 3rd Point

$$f(s_2) = \int_0^{s_2} \int_0^{s_2 - x_0} \int_0^{1 - s_2} {x_0}^{a_0} \, {x_1}^{a_1} \, (s_2 - x_0 - x_1)^{a_2} \, {x_3}^{a_3} \, (1 - s_2 - x_3)^{a_4} \, dx_3 dx_1 dx_0$$

$$f(s_2) = \int_0^{s_2} \int_0^{s_2 - x_0} {x_0}^{a_0} \, {x_1}^{a_1} \, (s_2 - x_0 - x_1)^{a_2} \, (1 - s_2)^{a_3 + a_4 + 1} \, \frac{a_3! \; a_4!}{(a_3 + a_4 + 1)!} \, dx_1 dx_0$$

$$f(s_2) = \int_0^{s_2} {x_0}^{a_0} \cdot (s_2 - x_0)^{a_1 + a_2 + 1} \, \frac{a_1! \; a_2!}{(a_1 + a_2 + 1)!} (1 - s_2)^{a_3 + a_4 + 1} \, \frac{a_3! \; a_4!}{(a_3 + a_4 + 1)!} \, dx_0$$

$$f(s_2) = {s_2}^{a_0 + a_1 + a_2 + 2} \, \frac{a_0! \; (a_1 + a_2 + 1)!}{(a_0 + a_1 + a_2 + 2)!} \, \frac{a_1! \; a_2!}{(a_1 + a_2 + 1)!} \, (1 - s_2)^{a_3 + a_4 + 1} \frac{a_3! \; a_4!}{(a_3 + a_4 + 1)!}$$

$$f(s_2) = {s_2}^{a_0 + a_1 + a_2 + 2} \, \frac{a_0! \; a_1! \; a_2!}{(a_0 + a_1 + a_2 + 2)!} \, (1 - s_2)^{a_3 + a_4 + 1} \, \frac{a_3! \; a_4!}{(a_3 + a_4 + 1)!}$$

A.3.4 Four ROC Points, 4th Point

$$f(s_3) = \int_0^{s_3} \int_0^{s_3 - x_0} \int_0^{s_3 - x_0 - x_1} {x_0}^{a_0} \, {x_1}^{a_1} \, {x_2}^{a_2} \, (s_3 - x_0 - x_1 - x_2)^{a_3} \, (1 - s_3)^{a_4} \, dx_2 dx_1 dx_0$$

$$f(s_3) = \int_0^{s_3} \int_0^{s_3 - x_0} {x_0}^{a_0} \, {x_1}^{a_1} \, (s_3 - x_0 - x_1)^{a_2 + a_3 + 1} \, \frac{a_2! \; a_3!}{(a_2 + a_3 + 1)!} \, (1 - s_3)^{a_4} \, dx_1 dx_0$$

$$f(s_3) = \int_0^{s_3} {x_0}^{a_0} \, (s_3 - x_0)^{a_1 + a_2 + a_3 + 2} \, \frac{a_1! \; (a_2 + a_3 + 1)!}{(a_1 + a_2 + a_3 + 2)!} \, \frac{a_2! \; a_3!}{(a_2 + a_3 + 1)!} \, (1 - s_3)^{a_4} \, dx_0$$

$$f(s_3) = {s_3}^{a_0 + a_1 + a_2 + a_3 + 3} \, \frac{a_0! \; (a_1 + a_2 + a_3 + 2)!}{(a_0 + a_1 + a_2 + a_3 + 3)!} \, \frac{a_1! \; a_2! \; a_3!}{(a_1 + a_2 + a_3 + 2)!} \, (1 - s_3)^{a_4}$$

$$f(s_3) = {s_3}^{a_0 + a_1 + a_2 + a_3 + 3} \, \frac{a_0! \; a_1! \; a_2! \; a_3!}{(a_0 + a_1 + a_2 + a_3 + 3)!} \, (1 - s_3)^{a_4}$$

B Results Tables

This appendix gives the result tables of all the Monte Carlo experiments discussed in Chapter 6. Full results are given for all simulations except those for nonparametric ROC points (Table B.2). For brevity, only the results for a one point ROC curve, the 5^{th} point of an 8 point ROC curve, and the 1^{st}, 8^{th}, and 16^{th} points of a 16 point ROC curve are given in this instance. The full table would be 61 pages long.

Selected rows from these tables are plotted as histograms in Chapter 6. These rows are highlighted in grey in the tables.

As an aid to interpretation, Table B.1 shows the distribution expected in theory if every experiment gave a perfect chi-squared distribution with 19 degrees of freedom. Because of the stochastic nature of Monte Carlo simulations correct results should not be expected to give perfect distributions, mearly a close approximation.

Example of Perfect Data											
			Theoretical Distribution of 10000 Chi-Squared Tests								
Pnts	Freq	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
1 of 1	1/2	1	6	465	2304	3280	2340	1087	378	107	26	7

Table B.1 Results expected in theory

Nonparametric ROC Points												
			Distribution of 200 Chi–Squared Tests × 2000 Experiments									
Pnts	Freq	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
1 of 1	1/2	1	0	7	51	43	56	28	10	5	0	0
1 of 1	1/2	2	0	13	47	54	45	23	14	4	0	0
1 of 1	1/2	4	0	8	43	77	44	17	10	1	0	0
1 of 1	1/2	8	0	9	46	67	47	20	11	0	0	0
1 of 1	1/2	16	0	10	46	58	54	24	6	2	0	0
1 of 1	1/2	32	0	14	55	59	42	23	3	3	0	1
1 of 1	1/2	64	0	11	48	68	47	16	4	5	0	1
1 of 1	1/2	128	1	10	42	70	53	19	4	1	0	0
1 of 1	1/2	256	0	10	42	65	54	22	4	2	1	0
1 of 1	1/2	512	0	14	41	76	37	24	7	1	0	0
1 of 1	1/2	1024	0	7	56	66	35	25	7	4	0	0
1 of 1	1/4	2	0	8	46	69	42	24	8	2	1	0
1 of 1	1/4	4	0	4	50	63	52	17	11	3	0	0
1 of 1	1/4	8	0	10	44	70	51	19	5	1	0	0
1 of 1	1/4	16	0	10	37	75	50	16	11	1	0	0
1 of 1	1/4	32	0	9	37	79	44	21	7	2	1	0
1 of 1	1/4	64	1	5	47	64	43	27	9	3	1	0
1 of 1	1/4	128	0	6	52	72	41	20	6	3	0	0
1 of 1	1/4	256	0	9	49	72	40	21	8	1	0	0
1 of 1	1/4	512	0	8	42	74	43	26	3	3	1	0
1 of 1	1/4	1024	0	8	46	63	49	25	5	4	0	0
1 of 1	1/8	4	0	6	44	68	45	27	4	3	3	0
1 of 1	1/8	8	0	8	46	66	50	24	4	2	0	0
1 of 1	1/8	16	0	10	43	59	45	30	9	3	1	0
1 of 1	1/8	32	1	6	43	68	44	27	8	1	1	1
1 of 1	1/8	64	0	11	44	68	42	19	11	5	0	0
1 of 1	1/8	128	1	12	55	52	48	21	8	2	1	0
1 of 1	1/8	256	0	14	48	64	52	16	3	2	1	0
1 of 1	1/8	512	1	9	46	73	41	20	7	3	0	0
1 of 1	1/8	1024	0	12	48	69	42	21	5	3	0	0
1 of 1	1/16	8	0	12	46	55	46	27	11	3	0	0
1 of 1	1/16	16	0	5	45	73	42	20	13	2	0	0
1 of 1	1/16	32	0	7	41	67	53	22	8	2	0	0

Table B.2 Nonparametric ROC Points

Nonparametric ROC Points												
			Distribution of 200 Chi–Squared Tests × 2000 Experiments									
Pnts	Freq	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
1 of 1	1/16	64	0	8	39	62	51	31	5	4	0	0
1 of 1	1/16	128	0	13	42	59	51	29	4	2	0	0
1 of 1	1/16	256	1	13	49	59	51	19	8	0	0	0
1 of 1	1/16	512	0	11	44	68	48	20	7	2	0	0
1 of 1	1/16	1024	0	9	49	72	44	20	4	1	1	0
1 of 1	1/32	16	0	9	41	70	53	20	4	2	1	0
1 of 1	1/32	32	0	8	42	73	48	16	8	4	1	0
1 of 1	1/32	64	0	8	45	76	40	22	7	2	0	0
1 of 1	1/32	128	0	11	48	67	44	19	6	4	0	1
1 of 1	1/32	256	0	9	54	73	43	15	4	1	1	0
1 of 1	1/32	512	0	11	40	74	44	20	7	0	1	3
1 of 1	1/32	1024	1	10	41	63	57	19	7	2	0	0
1 of 1	1/64	32	0	11	57	57	42	23	8	2	0	0
1 of 1	1/64	64	0	13	42	68	42	24	9	1	1	0
1 of 1	1/64	128	1	12	44	59	46	24	10	3	1	0
1 of 1	1/64	256	0	7	46	65	44	29	5	2	2	0
1 of 1	1/64	512	0	12	47	62	50	15	12	2	0	0
1 of 1	1/64	1024	0	6	50	67	47	19	8	3	0	0
1 of 1	1/128	64	1	10	30	77	45	30	5	2	0	0
1 of 1	1/128	128	0	8	48	66	44	18	12	3	1	0
1 of 1	1/128	256	1	3	56	60	45	27	4	3	1	0
1 of 1	1/128	512	0	9	43	73	49	16	6	4	0	0
1 of 1	1/128	1024	0	12	45	64	42	19	13	4	1	0
1 of 1	1/256	128	0	9	51	69	46	17	7	1	0	0
1 of 1	1/256	256	0	10	41	71	45	25	6	2	0	0
1 of 1	1/256	512	0	6	35	73	47	25	11	2	1	0
1 of 1	1/256	1024	0	8	57	64	48	15	5	3	0	0
1 of 1	1/512	256	1	8	51	70	37	26	5	2	0	0
1 of 1	1/512	512	0	10	42	64	48	24	8	2	0	2
1 of 1	1/512	1024	0	9	52	61	40	27	8	3	0	0
1 of 1	1/1024	512	0	15	35	75	45	22	5	2	0	1
1 of 1	1/1024	1024	1	10	52	61	51	14	8	3	0	0
1 of 1	1/2048	1024	0	13	50	67	44	20	3	2	0	1

Table B.2 ... continued ...

Nonparametric ROC Points												
			Distribution of 200 Chi–Squared Tests × 2000 Experiments									
Pnts	Freq	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
5 of 8	1/2	1	0	13	58	57	34	26	10	0	1	1
5 of 8	1/2	2	0	3	43	69	44	31	9	1	0	0
5 of 8	1/2	4	0	11	49	57	53	17	10	3	0	0
5 of 8	1/2	8	0	8	39	72	53	23	4	0	0	1
5 of 8	1/2	16	0	12	46	63	47	22	5	4	1	0
5 of 8	1/2	32	0	13	50	70	42	20	4	1	0	0
5 of 8	1/2	64	0	10	52	69	41	18	5	5	0	0
5 of 8	1/2	128	0	13	45	65	47	25	3	2	0	0
5 of 8	1/2	256	0	9	45	62	46	27	9	2	0	0
5 of 8	1/2	512	0	9	43	60	48	27	8	2	2	1
5 of 8	1/2	1024	0	16	51	54	44	23	7	5	0	0
5 of 8	1/4	2	0	11	48	55	56	22	7	1	0	0
5 of 8	1/4	4	0	8	53	64	42	20	11	2	0	0
5 of 8	1/4	8	0	8	43	68	51	16	12	2	0	0
5 of 8	1/4	16	0	13	43	63	46	21	10	2	2	0
5 of 8	1/4	32	0	8	50	63	45	25	5	3	0	1
5 of 8	1/4	64	0	7	53	67	51	15	5	1	1	0
5 of 8	1/4	128	0	12	46	63	35	28	8	8	0	0
5 of 8	1/4	256	0	8	43	78	39	21	7	2	1	1
5 of 8	1/4	512	0	10	42	65	39	32	6	6	0	0
5 of 8	1/4	1024	0	7	49	61	48	26	8	0	1	0
5 of 8	1/8	4	0	8	43	71	36	29	8	5	0	0
5 of 8	1/8	8	0	16	41	72	49	16	3	1	2	0
5 of 8	1/8	16	0	12	40	88	37	14	7	2	0	0
5 of 8	1/8	32	0	7	39	64	52	28	7	3	0	0
5 of 8	1/8	64	0	7	45	71	52	19	4	2	0	0
5 of 8	1/8	128	0	9	54	61	50	19	5	2	0	0
5 of 8	1/8	256	0	9	45	59	50	29	4	3	1	0
5 of 8	1/8	512	0	11	43	68	40	28	9	1	0	0
5 of 8	1/8	1024	1	3	55	63	45	26	6	1	0	0
5 of 8	1/16	8	0	10	54	60	43	24	7	2	0	0
5 of 8	1/16	16	0	15	50	60	44	24	6	1	0	0
5 of 8	1/16	32	0	14	44	53	52	30	6	0	1	0

Table B.2 ... continued ...

Nonparametric ROC Points												
			Distribution of 200 Chi-Squared Tests × 2000 Experiments									
Pnts	Freq	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
5 of 8	1/16	64	0	14	41	61	49	24	10	0	1	0
5 of 8	1/16	128	0	10	44	69	50	16	6	4	1	0
5 of 8	1/16	256	0	2	50	59	58	23	4	3	0	1
5 of 8	1/16	512	0	8	43	70	51	21	5	2	0	0
5 of 8	1/16	1024	0	10	44	64	53	20	6	2	1	0
5 of 8	1/32	16	0	9	48	60	46	21	12	2	2	0
5 of 8	1/32	32	0	12	50	60	46	23	7	0	2	0
5 of 8	1/32	64	0	9	52	68	41	17	10	2	1	0
5 of 8	1/32	128	0	12	45	61	48	26	5	3	0	0
5 of 8	1/32	256	0	7	42	65	50	26	7	3	0	0
5 of 8	1/32	512	0	7	49	65	43	25	11	0	0	0
5 of 8	1/32	1024	0	3	49	63	48	26	6	3	2	0
5 of 8	1/64	32	0	6	48	68	47	25	5	1	0	0
5 of 8	1/64	64	1	12	53	65	42	19	6	1	1	0
5 of 8	1/64	128	0	6	43	65	49	32	0	2	2	1
5 of 8	1/64	256	0	14	37	62	50	25	9	2	0	1
5 of 8	1/64	512	0	5	53	60	43	23	13	3	0	0
5 of 8	1/64	1024	0	12	42	74	45	17	6	4	0	0
5 of 8	1/128	64	0	10	49	63	47	13	16	1	0	1
5 of 8	1/128	128	1	16	34	70	47	26	5	1	0	0
5 of 8	1/128	256	0	12	47	61	49	17	11	2	1	0
5 of 8	1/128	512	0	4	35	69	51	30	6	5	0	0
5 of 8	1/128	1024	0	5	47	69	47	26	4	1	1	0
5 of 8	1/256	128	0	6	50	72	41	20	10	0	1	0
5 of 8	1/256	256	1	10	47	69	36	27	10	0	0	0
5 of 8	1/256	512	0	9	56	64	36	23	8	3	1	0
5 of 8	1/256	1024	0	7	49	65	52	19	6	0	2	0
5 of 8	1/512	256	0	7	48	62	45	26	7	5	0	0
5 of 8	1/512	512	0	15	35	64	60	19	6	1	0	0
5 of 8	1/512	1024	0	13	42	62	53	22	7	1	0	0
5 of 8	1/1024	512	0	11	45	68	42	19	12	1	2	0
5 of 8	1/1024	1024	0	11	40	74	34	29	11	0	1	0
5 of 8	1/2048	1024	0	10	38	52	61	30	7	2	0	0

Table B.2 ... continued ...

Nonparametric ROC Points												
			Distribution of 200 Chi-Squared Tests × 2000 Experiments									
Pnts	Freq	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
1 of 16	1/2	1	0	7	57	64	40	19	9	1	1	2
1 of 16	1/2	2	0	8	40	67	46	28	9	2	0	0
1 of 16	1/2	4	0	11	49	64	46	23	6	1	0	0
1 of 16	1/2	8	0	11	49	55	50	23	9	2	1	0
1 of 16	1/2	16	0	3	46	77	44	20	8	2	0	0
1 of 16	1/2	32	0	14	50	61	48	16	7	2	2	0
1 of 16	1/2	64	0	8	46	77	43	19	4	1	1	1
1 of 16	1/2	128	0	12	48	72	39	19	6	3	1	0
1 of 16	1/2	256	0	10	49	61	54	20	3	3	0	0
1 of 16	1/2	512	0	9	47	69	42	21	11	1	0	0
1 of 16	1/2	1024	0	9	37	75	44	26	6	1	2	0
1 of 16	1/4	2	0	11	58	58	38	27	8	0	0	0
1 of 16	1/4	4	0	8	36	70	38	32	11	5	0	0
1 of 16	1/4	8	0	3	58	65	44	17	9	3	1	0
1 of 16	1/4	16	0	10	44	55	52	25	12	2	0	0
1 of 16	1/4	32	0	11	49	62	53	17	8	0	0	0
1 of 16	1/4	64	0	8	45	69	45	21	8	3	1	0
1 of 16	1/4	128	0	12	53	64	38	18	10	4	0	1
1 of 16	1/4	256	0	14	34	68	51	26	6	1	0	0
1 of 16	1/4	512	0	7	56	59	42	19	8	8	1	0
1 of 16	1/4	1024	0	6	37	64	57	24	9	2	1	0
1 of 16	1/8	4	0	9	39	66	53	26	6	1	0	0
1 of 16	1/8	8	0	9	46	63	54	19	6	2	0	1
1 of 16	1/8	16	0	11	49	60	43	29	7	1	0	0
1 of 16	1/8	32	0	11	40	67	58	21	3	0	0	0
1 of 16	1/8	64	0	6	49	79	34	19	10	3	0	0
1 of 16	1/8	128	0	7	44	64	48	26	11	0	0	0
1 of 16	1/8	256	0	12	47	63	50	18	5	4	0	1
1 of 16	1/8	512	0	11	50	62	51	16	4	6	0	0
1 of 16	1/8	1024	0	9	42	59	50	24	14	0	1	1
1 of 16	1/16	8	0	6	39	73	48	27	6	1	0	0
1 of 16	1/16	16	0	9	59	55	47	18	11	0	1	0
1 of 16	1/16	32	0	12	23	70	55	24	12	4	0	0

Table B.2 ... continued ...

Nonparametric ROC Points												
			Distribution of 200 Chi–Squared Tests × 2000 Experiments									
Pnts	Freq	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
1 of 16	1/16	64	0	13	41	70	43	27	5	1	0	0
1 of 16	1/16	128	0	8	42	73	47	18	5	6	1	0
1 of 16	1/16	256	1	10	41	74	48	15	9	2	0	0
1 of 16	1/16	512	0	5	47	72	53	12	8	2	1	0
1 of 16	1/16	1024	0	8	49	62	42	27	10	1	1	0
1 of 16	1/32	16	0	15	46	62	40	27	8	2	0	0
1 of 16	1/32	32	0	8	50	66	54	11	7	4	0	0
1 of 16	1/32	64	0	5	52	65	42	23	10	3	0	0
1 of 16	1/32	128	0	10	31	80	53	19	5	1	1	0
1 of 16	1/32	256	0	8	39	61	59	22	8	3	0	0
1 of 16	1/32	512	1	9	37	67	52	23	8	2	1	0
1 of 16	1/32	1024	0	9	35	63	46	35	10	0	2	0
1 of 16	1/64	32	0	11	36	64	53	29	3	3	1	0
1 of 16	1/64	64	0	4	54	66	43	21	12	0	0	0
1 of 16	1/64	128	0	8	48	59	47	22	12	3	1	0
1 of 16	1/64	256	0	15	50	62	44	18	8	2	1	0
1 of 16	1/64	512	1	8	58	64	43	17	6	1	2	0
1 of 16	1/64	1024	0	8	53	61	48	14	8	8	0	0
1 of 16	1/128	64	0	10	43	60	49	23	11	1	2	1
1 of 16	1/128	128	0	2	39	70	43	32	6	5	1	2
1 of 16	1/128	256	0	11	43	72	45	14	11	4	0	0
1 of 16	1/128	512	0	9	50	62	46	20	10	2	1	0
1 of 16	1/128	1024	0	7	55	64	54	12	6	1	1	0
1 of 16	1/256	128	0	15	54	54	45	23	6	3	0	0
1 of 16	1/256	256	0	9	42	75	40	17	11	5	1	0
1 of 16	1/256	512	0	10	44	63	43	29	6	3	1	1
1 of 16	1/256	1024	0	8	51	70	38	19	9	3	1	1
1 of 16	1/512	256	0	14	47	62	42	23	9	2	0	1
1 of 16	1/512	512	1	11	44	58	48	25	5	7	1	0
1 of 16	1/512	1024	1	11	48	51	57	20	8	4	0	0
1 of 16	1/1024	512	0	7	45	81	41	20	5	1	0	0
1 of 16	1/1024	1024	0	8	36	76	46	21	9	4	0	0
1 of 16	1/2048	1024	0	10	36	62	55	29	7	1	0	0

Table B.2 ... continued ...

Nonparametric ROC Points												
			Distribution of 200 Chi–Squared Tests × 2000 Experiments									
Pnts	Freq	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
8 of 16	1/2	1	0	11	50	60	39	32	6	2	0	0
8 of 16	1/2	2	0	12	42	73	48	18	5	2	0	0
8 of 16	1/2	4	0	10	45	78	42	18	5	2	0	0
8 of 16	1/2	8	0	9	43	70	41	23	10	4	0	0
8 of 16	1/2	16	0	11	41	80	40	17	7	4	0	0
8 of 16	1/2	32	0	10	50	73	37	23	6	0	1	0
8 of 16	1/2	64	0	6	48	75	43	22	2	2	2	0
8 of 16	1/2	128	0	11	41	68	42	19	16	3	0	0
8 of 16	1/2	256	0	9	45	71	44	21	9	1	0	0
8 of 16	1/2	512	0	5	46	72	53	19	3	2	0	0
8 of 16	1/2	1024	1	16	42	61	51	20	8	0	1	0
8 of 16	1/4	2	0	8	47	71	40	26	6	2	0	0
8 of 16	1/4	4	0	7	46	62	55	20	9	1	0	0
8 of 16	1/4	8	0	13	44	73	48	12	9	1	0	0
8 of 16	1/4	16	0	11	40	72	44	23	6	3	0	1
8 of 16	1/4	32	0	8	43	71	51	17	9	1	0	0
8 of 16	1/4	64	0	7	46	63	50	28	4	2	0	0
8 of 16	1/4	128	0	10	41	66	46	24	10	2	1	0
8 of 16	1/4	256	0	8	38	77	54	13	7	3	0	0
8 of 16	1/4	512	0	5	51	67	44	25	8	0	0	0
8 of 16	1/4	1024	0	7	42	59	50	30	10	2	0	0
8 of 16	1/8	4	1	13	44	59	48	29	2	3	0	1
8 of 16	1/8	8	0	10	46	64	44	25	9	1	1	0
8 of 16	1/8	16	1	10	43	58	45	29	11	2	0	1
8 of 16	1/8	32	0	8	46	72	42	22	9	1	0	0
8 of 16	1/8	64	0	13	39	69	53	19	7	0	0	0
8 of 16	1/8	128	0	11	42	73	53	16	4	1	0	0
8 of 16	1/8	256	0	12	39	58	56	22	10	2	1	0
8 of 16	1/8	512	2	11	36	69	44	26	8	2	2	0
8 of 16	1/8	1024	0	6	54	61	41	30	7	1	0	0
8 of 16	1/16	8	0	13	58	62	41	15	7	3	1	0
8 of 16	1/16	16	0	15	55	57	49	21	3	0	0	0
8 of 16	1/16	32	0	13	40	70	51	17	8	1	0	0

Table B.2 ... continued ...

Nonparametric ROC Points												
			Distribution of 200 Chi-Squared Tests × 2000 Experiments									
Pnts	Freq	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
8 of 16	1/16	64	0	6	54	67	47	17	8	1	0	0
8 of 16	1/16	128	0	10	42	66	45	28	8	1	0	0
8 of 16	1/16	256	0	11	46	63	48	21	10	1	0	0
8 of 16	1/16	512	1	7	55	67	47	17	4	2	0	0
8 of 16	1/16	1024	0	17	49	59	49	18	5	2	1	0
8 of 16	1/32	16	0	13	50	66	40	25	3	2	1	0
8 of 16	1/32	32	0	12	54	60	43	19	8	3	1	0
8 of 16	1/32	64	0	5	37	64	53	28	11	2	0	0
8 of 16	1/32	128	1	8	41	72	47	24	4	1	2	0
8 of 16	1/32	256	0	8	45	62	41	32	8	3	1	0
8 of 16	1/32	512	0	12	46	65	37	28	10	2	0	0
8 of 16	1/32	1024	1	15	44	53	44	30	9	4	0	0
8 of 16	1/64	32	2	9	42	56	48	33	6	3	1	0
8 of 16	1/64	64	0	8	44	75	47	23	3	0	0	0
8 of 16	1/64	128	0	11	38	68	47	31	5	0	0	0
8 of 16	1/64	256	0	8	45	66	45	20	12	4	0	0
8 of 16	1/64	512	0	10	41	70	47	25	6	1	0	0
8 of 16	1/64	1024	0	13	48	57	45	19	11	6	0	1
8 of 16	1/128	64	0	9	53	61	39	29	7	2	0	0
8 of 16	1/128	128	0	4	45	77	48	19	6	1	0	0
8 of 16	1/128	256	0	10	40	60	52	28	6	4	0	0
8 of 16	1/128	512	0	6	49	64	55	18	6	1	1	0
8 of 16	1/128	1024	1	9	49	72	39	21	6	2	1	0
8 of 16	1/256	128	0	10	52	64	42	25	6	0	1	0
8 of 16	1/256	256	0	10	42	71	47	20	8	2	0	0
8 of 16	1/256	512	0	9	40	69	54	18	8	1	1	0
8 of 16	1/256	1024	0	17	36	66	45	23	10	3	0	0
8 of 16	1/512	256	0	9	46	59	52	26	6	1	1	0
8 of 16	1/512	512	0	9	35	57	58	23	12	5	1	0
8 of 16	1/512	1024	0	15	42	69	41	19	8	4	2	0
8 of 16	1/1024	512	1	9	48	66	49	22	3	2	0	0
8 of 16	1/1024	1024	0	6	43	82	41	18	7	3	0	0
8 of 16	1/2048	1024	0	15	54	54	44	25	6	2	0	0

Table B.2 ... continued ...

Nonparametric ROC Points												
			Distribution of 200 Chi–Squared Tests × 2000 Experiments									
Pnts	Freq	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
16 of 16	1/2	1	0	14	34	69	51	25	6	0	1	0
16 of 16	1/2	2	0	15	38	67	45	24	9	2	0	0
16 of 16	1/2	4	1	10	43	73	46	19	7	1	0	0
16 of 16	1/2	8	0	11	44	67	51	18	4	5	0	0
16 of 16	1/2	16	0	7	52	65	49	23	4	0	0	0
16 of 16	1/2	32	0	5	53	69	48	18	5	2	0	0
16 of 16	1/2	64	1	8	45	65	47	20	12	2	0	0
16 of 16	1/2	128	0	11	50	66	44	20	7	1	1	0
16 of 16	1/2	256	0	16	47	51	53	19	9	4	1	0
16 of 16	1/2	512	0	11	43	79	41	16	6	4	0	0
16 of 16	1/2	1024	0	8	33	64	63	23	6	3	0	0
16 of 16	1/4	2	1	16	26	68	47	30	8	4	0	0
16 of 16	1/4	4	0	11	40	71	51	20	4	1	0	2
16 of 16	1/4	8	1	6	46	65	48	21	12	1	0	0
16 of 16	1/4	16	0	14	42	68	50	20	3	2	1	0
16 of 16	1/4	32	0	13	52	61	45	18	7	3	0	1
16 of 16	1/4	64	0	5	42	70	53	20	7	3	0	0
16 of 16	1/4	128	0	13	48	59	53	17	8	1	1	0
16 of 16	1/4	256	0	9	47	50	59	21	9	5	0	0
16 of 16	1/4	512	0	7	54	64	49	15	9	1	1	0
16 of 16	1/4	1024	0	6	38	68	53	15	13	6	1	0
16 of 16	1/8	4	0	9	40	57	55	24	12	3	0	0
16 of 16	1/8	8	0	12	49	58	44	29	6	2	0	0
16 of 16	1/8	16	0	13	42	67	46	14	13	5	0	0
16 of 16	1/8	32	0	4	38	79	50	23	6	0	0	0
16 of 16	1/8	64	0	11	53	69	41	20	4	2	0	0
16 of 16	1/8	128	0	6	45	70	43	29	4	1	2	0
16 of 16	1/8	256	0	11	53	61	42	21	10	0	1	1
16 of 16	1/8	512	0	9	46	67	45	24	7	2	0	0
16 of 16	1/8	1024	0	7	33	68	49	23	13	4	3	0
16 of 16	1/16	8	0	12	56	62	45	15	8	1	1	0
16 of 16	1/16	16	0	12	49	62	39	23	12	1	2	0
16 of 16	1/16	32	0	7	50	63	41	21	16	1	1	0

Table B.2 ... continued ...

Nonparametric ROC Points												
			Distribution of 200 Chi–Squared Tests × 2000 Experiments									
Pnts	Freq	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
16 of 16	1/16	64	1	12	51	65	40	16	10	2	3	0
16 of 16	1/16	128	0	4	43	69	51	19	7	4	3	0
16 of 16	1/16	256	0	12	34	73	50	21	7	1	2	0
16 of 16	1/16	512	0	16	36	69	37	23	13	6	0	0
16 of 16	1/16	1024	0	5	40	62	55	26	6	6	0	0
16 of 16	1/32	16	0	9	56	62	47	22	2	0	1	1
16 of 16	1/32	32	0	14	46	54	54	18	9	4	0	1
16 of 16	1/32	64	1	13	48	60	43	26	8	1	0	0
16 of 16	1/32	128	0	9	45	75	48	18	5	0	0	0
16 of 16	1/32	256	0	8	49	77	44	14	4	4	0	0
16 of 16	1/32	512	0	14	46	73	38	24	4	1	0	0
16 of 16	1/32	1024	1	6	34	65	58	22	9	4	1	0
16 of 16	1/64	32	0	11	48	69	45	18	6	3	0	0
16 of 16	1/64	64	0	11	42	73	43	21	9	1	0	0
16 of 16	1/64	128	0	15	51	64	35	20	11	2	2	0
16 of 16	1/64	256	1	12	43	73	46	16	5	3	0	1
16 of 16	1/64	512	0	15	42	57	48	28	4	3	3	0
16 of 16	1/64	1024	0	4	40	78	35	25	12	6	0	0
16 of 16	1/128	64	0	5	50	64	50	22	7	0	1	1
16 of 16	1/128	128	0	9	49	61	45	27	9	0	0	0
16 of 16	1/128	256	0	8	44	65	44	21	12	4	2	0
16 of 16	1/128	512	0	10	46	65	38	24	13	3	1	0
16 of 16	1/128	1024	0	9	47	59	43	24	13	4	0	1
16 of 16	1/256	128	0	8	45	62	51	29	4	1	0	0
16 of 16	1/256	256	0	11	39	67	49	22	9	3	0	0
16 of 16	1/256	512	0	8	45	68	49	20	6	3	1	0
16 of 16	1/256	1024	0	11	42	64	49	19	11	3	1	0
16 of 16	1/512	256	0	9	48	66	47	21	7	2	0	0
16 of 16	1/512	512	0	8	47	56	53	28	8	0	0	0
16 of 16	1/512	1024	0	9	47	66	40	26	9	1	2	0
16 of 16	1/1024	512	0	5	52	60	47	27	5	3	1	0
16 of 16	1/1024	1024	0	11	42	66	47	22	8	2	2	0
16 of 16	1/2048	1024	0	6	46	69	46	15	14	4	0	0

Table B.2 ... end.

Nonparametric ROC Point for Fixed Population											
		Distribution of 200 Chi–Squared Tests × 2000 Experiments									
Freq	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
1/2	1	0	0	0	0	0	0	0	0	0	200
1/2	2	0	0	0	0	0	0	0	0	0	200
1/2	4	0	0	0	0	0	0	0	0	0	200
1/2	8	0	0	0	0	0	0	0	0	0	200
1/2	16	0	0	0	0	0	0	0	0	0	200
1/2	32	0	0	0	0	0	0	0	0	0	200
1/2	64	0	0	0	0	0	0	0	0	0	200
1/2	128	0	1	12	33	44	42	33	17	9	9
1/2	256	0	2	27	48	63	38	12	5	3	2
1/2	512	0	5	44	62	53	21	9	4	2	0
1/2	1024	0	10	41	63	39	27	11	8	1	0
1/4	2	0	0	0	0	0	0	0	0	0	200
1/4	4	0	0	0	0	0	0	0	0	0	200
1/4	8	0	0	0	0	0	0	0	0	0	200
1/4	16	0	0	0	0	0	0	0	0	0	200
1/4	32	0	0	0	0	0	0	0	0	0	200
1/4	64	0	0	0	0	0	0	0	0	0	200
1/4	128	0	1	4	19	41	44	37	19	18	17
1/4	256	0	0	11	34	59	43	29	15	6	3
1/4	512	1	7	30	67	47	28	17	3	0	0
1/4	1024	0	7	45	65	42	31	7	3	0	0
1/8	4	0	0	0	0	0	0	0	0	0	200
1/8	8	0	0	0	0	0	0	0	0	0	200
1/8	16	0	0	0	0	0	0	0	0	0	200
1/8	32	0	0	0	0	0	0	0	0	0	200
1/8	64	0	0	0	0	0	0	0	0	0	200
1/8	128	0	0	0	4	17	35	40	39	32	33
1/8	256	0	4	19	43	51	41	23	14	2	3
1/8	512	0	3	27	50	51	39	20	8	2	0
1/8	1024	0	4	42	64	52	27	9	2	0	0
1/16	8	0	0	0	0	0	0	0	0	0	200
1/16	16	0	0	0	0	0	0	0	0	0	200
1/16	32	0	0	0	0	0	0	0	0	0	200

Table B.3 Nonparametric ROC Point for Fixed Population

Nonparametric ROC Point for Fixed Population											
		Distribution of 200 Chi-Squared Tests × 2000 Experiments									
Freq	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
1/16	64	0	0	0	0	0	0	0	0	0	200
1/16	128	0	0	0	0	0	0	0	0	0	200
1/16	256	0	0	3	12	30	38	45	26	21	25
1/16	512	0	3	22	59	64	33	13	4	1	1
1/16	1024	0	4	31	65	56	31	9	2	1	1
1/32	16	0	0	0	0	0	0	0	0	0	200
1/32	32	0	0	0	0	0	0	0	0	0	200
1/32	64	0	0	0	0	0	0	0	0	0	200
1/32	128	0	0	0	0	0	0	0	0	0	200
1/32	256	0	0	0	0	0	0	5	14	20	161
1/32	512	0	1	10	48	43	43	28	17	5	5
1/32	1024	0	9	33	47	53	41	11	3	3	0
1/64	32	0	0	0	0	0	0	0	0	0	200
1/64	64	0	0	0	0	0	0	0	0	0	200
1/64	128	0	0	0	0	0	0	0	0	0	200
1/64	256	0	0	0	0	0	0	0	0	0	200
1/64	512	0	0	0	0	4	7	17	25	26	121
1/64	1024	0	1	10	36	50	45	29	17	5	7
1/128	64	0	0	0	0	0	0	0	0	0	200
1/128	128	0	0	0	0	0	0	0	0	0	200
1/128	256	0	0	0	0	0	0	0	0	0	200
1/128	512	0	0	0	0	0	0	0	0	0	200
1/128	1024	0	0	0	1	5	0	19	24	21	130
1/256	128	0	0	0	0	0	0	0	0	0	200
1/256	256	0	0	0	0	0	0	0	0	0	200
1/256	512	0	0	0	0	0	0	0	0	0	200
1/256	1024	0	0	0	0	0	0	0	0	0	200
1/512	256	0	0	0	0	0	0	0	0	0	200
1/512	512	0	0	0	0	0	0	0	0	0	200
1/512	1024	0	0	0	0	0	0	0	0	0	200
1/1024	512	0	0	0	0	0	0	0	0	0	200
1/1024	1024	0	0	0	0	0	0	0	0	0	200
1/2048	1024	0	0	0	0	0	0	0	0	0	200

Table B.3 ... end

Nonparametric ROC Point Comparison													
Exp 1		Exp 2		Distribution of 200 Chi-Squared Tests × 2000 Experiments									
Freq	Case	Freq	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
1/2	1	1/2	1	0	10	44	64	53	22	3	3	1	0
1/2	1	1/2	4	0	12	54	64	40	19	8	2	1	0
1/2	1	1/2	16	0	12	57	59	45	14	11	2	0	0
1/2	1	1/2	64	0	7	56	62	38	21	13	3	0	0
1/2	1	1/2	256	0	9	46	65	47	24	6	2	1	0
1/2	1	1/2	1024	1	7	48	70	47	20	4	3	0	0
1/2	4	1/2	4	1	7	48	70	47	20	4	3	0	0
1/2	4	1/2	16	1	9	53	55	46	31	4	1	0	0
1/2	4	1/2	64	0	9	42	67	57	15	5	3	1	1
1/2	4	1/2	256	0	7	51	70	42	20	9	1	0	0
1/2	4	1/2	1024	0	10	51	63	44	22	7	2	1	0
1/2	16	1/2	16	1	8	55	60	49	20	4	2	1	0
1/2	16	1/2	64	0	9	52	66	40	24	5	3	1	0
1/2	16	1/2	256	0	9	40	65	53	17	14	2	0	0
1/2	16	1/2	1024	0	16	47	63	40	21	9	2	2	0
1/2	64	1/2	64	0	13	42	54	59	22	6	3	1	0
1/2	64	1/2	256	0	8	36	66	49	24	13	3	1	0
1/2	64	1/2	1024	0	15	46	61	40	28	9	0	1	0
1/2	256	1/2	256	0	11	48	69	40	22	6	2	2	0
1/2	256	1/2	1024	0	7	49	66	49	23	5	1	0	0
1/2	1024	1/2	1024	0	6	50	72	49	16	4	3	0	0
1/8	1	1/8	1	1	9	51	72	37	22	4	3	1	0
1/8	1	1/8	4	0	6	50	72	49	16	4	3	0	0
1/8	1	1/8	16	1	10	56	54	47	26	3	3	0	0
1/8	1	1/8	64	0	16	43	57	53	18	9	3	1	0
1/8	1	1/8	256	0	7	51	59	47	26	3	6	1	0
1/8	1	1/8	1024	0	12	47	64	53	18	3	1	2	0
1/8	4	1/8	4	0	12	47	64	53	18	3	1	2	0
1/8	4	1/8	16	0	12	39	59	61	22	7	0	0	0
1/8	4	1/8	64	0	9	38	60	53	27	11	2	0	0
1/8	4	1/8	256	0	6	50	62	43	22	13	4	0	0
1/8	4	1/8	1024	0	8	50	66	36	19	11	6	2	2
1/8	16	1/8	16	0	11	37	79	38	28	6	1	0	0

Table B.4 Nonparametric ROC Point Comparison

Nonparametric ROC Point Comparison													
Exp 1		Exp 2		Distribution of 200 Chi-Squared Tests × 2000 Experiments									
Freq	Case	Freq	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
1/8	16	1/8	64	0	10	49	59	54	17	7	2	1	1
1/8	16	1/8	256	0	12	46	71	49	15	5	2	0	0
1/8	16	1/8	1024	1	6	42	80	40	25	4	2	0	0
1/8	64	1/8	64	0	7	57	48	55	22	8	3	0	0
1/8	64	1/8	256	0	7	52	74	38	17	9	1	2	0
1/8	64	1/8	1024	0	9	50	53	51	28	6	3	0	0
1/8	256	1/8	256	0	9	40	67	55	22	4	3	0	0
1/8	256	1/8	1024	0	5	44	73	49	18	9	2	0	0
1/8	1024	1/8	1024	0	6	54	75	37	18	8	1	1	0
1/2	1	1/8	1	0	12	49	64	44	20	3	7	1	0
1/2	1	1/8	4	0	6	54	75	37	18	8	1	1	0
1/2	1	1/8	16	0	12	44	64	54	19	7	0	0	0
1/2	1	1/8	64	0	10	48	76	41	18	6	1	0	0
1/2	1	1/8	256	0	7	58	47	46	32	8	2	0	0
1/2	1	1/8	1024	0	9	49	71	42	14	4	9	2	0
1/2	4	1/8	4	0	9	49	71	42	14	4	9	2	0
1/2	4	1/8	16	0	7	43	63	44	34	6	2	1	0
1/2	4	1/8	64	0	7	43	55	63	22	7	3	0	0
1/2	4	1/8	256	0	10	47	57	57	20	6	3	0	0
1/2	4	1/8	1024	0	5	49	63	46	29	6	1	1	0
1/2	16	1/8	16	0	10	36	68	55	18	8	3	1	1
1/2	16	1/8	64	0	6	39	71	54	23	6	1	0	0
1/2	16	1/8	256	0	8	44	72	41	25	7	2	1	0
1/2	16	1/8	1024	0	11	51	64	42	21	10	0	1	0
1/2	64	1/8	64	0	8	43	68	48	26	4	2	1	0
1/2	64	1/8	256	0	10	44	70	43	25	5	3	0	0
1/2	64	1/8	1024	0	9	37	71	50	26	7	0	0	0
1/2	256	1/8	256	0	4	48	69	44	26	7	2	0	0
1/2	256	1/8	1024	0	8	42	67	53	21	9	0	0	0
1/2	1024	1/8	1024	0	13	38	66	51	15	13	3	1	0
1/8	1	7/8	1	0	5	45	71	34	22	21	1	1	0
1/8	1	7/8	4	0	13	38	66	51	15	13	3	1	0
1/8	1	7/8	16	0	7	40	69	49	23	10	2	0	0

Table B.4 ... continued ...

Nonparametric ROC Point Comparison													
Exp 1		Exp 2		Distribution of 200 Chi-Squared Tests × 2000 Experiments									
Freq	Case	Freq	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
1/8	1	7/8	64	0	12	44	67	39	27	10	1	0	0
1/8	1	7/8	256	1	9	48	61	55	17	6	2	1	0
1/8	1	7/8	1024	0	7	38	69	47	27	8	4	0	0
1/8	4	7/8	4	0	7	38	69	47	27	8	4	0	0
1/8	4	7/8	16	0	7	42	63	51	27	10	0	0	0
1/8	4	7/8	64	0	8	52	61	50	19	8	1	1	0
1/8	4	7/8	256	0	6	39	62	56	24	9	2	1	1
1/8	4	7/8	1024	0	8	42	72	53	15	9	1	0	0
1/8	16	7/8	16	0	6	44	77	47	16	8	1	1	0
1/8	16	7/8	64	0	10	40	66	52	19	10	3	0	0
1/8	16	7/8	256	0	12	48	63	39	30	6	1	1	0
1/8	16	7/8	1024	0	10	54	52	49	27	6	1	1	0
1/8	64	7/8	64	0	8	46	66	45	24	6	5	0	0
1/8	64	7/8	256	0	8	51	56	51	20	11	3	0	0
1/8	64	7/8	1024	1	14	49	63	44	17	11	1	0	0
1/8	256	7/8	256	1	10	35	74	45	26	9	0	0	0
1/8	256	7/8	1024	0	12	44	60	48	22	9	4	1	0
1/8	1024	7/8	1024	0	8	63	48	45	23	7	5	1	0

Table B.4 ... end

Nonparametric AUC												
			Distribution of 1 Chi–Squared Tests × 1000 Experiments									
Pnts	Freq	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
4	1/2	8	0	0	0	0	1	0	0	0	0	0
4	1/2	16	0	0	0	0	1	0	0	0	0	0
4	1/2	32	0	0	0	0	1	0	0	0	0	0
4	1/2	64	0	0	0	1	0	0	0	0	0	0
4	1/2	128	0	0	0	1	0	0	0	0	0	0
4	1/4	16	0	0	0	0	1	0	0	0	0	0
4	1/4	32	0	0	1	0	0	0	0	0	0	0
4	1/4	64	0	0	1	0	0	0	0	0	0	0
4	1/4	128	0	0	0	1	0	0	0	0	0	0
4	1/8	32	0	0	0	0	0	1	0	0	0	0
4	1/8	64	0	0	0	0	1	0	0	0	0	0
4	1/8	128	0	0	1	0	0	0	0	0	0	0
8	1/2	8	0	0	1	0	0	0	0	0	0	0
8	1/2	16	0	0	0	0	1	0	0	0	0	0
8	1/2	32	0	0	0	0	1	0	0	0	0	0
8	1/2	64	0	0	0	0	1	0	0	0	0	0
8	1/2	128	0	0	0	0	1	0	0	0	0	0
8	1/4	16	0	0	1	0	0	0	0	0	0	0
8	1/4	32	0	0	0	0	0	1	0	0	0	0
8	1/4	64	0	0	0	1	0	0	0	0	0	0
8	1/4	128	0	0	1	0	0	0	0	0	0	0
8	1/8	32	0	0	0	0	1	0	0	0	0	0
8	1/8	64	0	0	0	0	1	0	0	0	0	0
8	1/8	128	0	0	1	0	0	0	0	0	0	0
16	1/2	8	0	0	0	1	0	0	0	0	0	0
16	1/2	16	0	0	0	0	0	0	1	0	0	0
16	1/2	32	0	0	1	0	0	0	0	0	0	0
16	1/2	64	0	1	0	0	0	0	0	0	0	0
16	1/2	128	0	0	1	0	0	0	0	0	0	0
16	1/4	16	0	0	1	0	0	0	0	0	0	0
16	1/4	32	0	0	0	1	0	0	0	0	0	0
16	1/4	64	0	0	0	1	0	0	0	0	0	0
16	1/4	128	0	0	0	1	0	0	0	0	0	0

Table B.5 Nonparametric AUC

Nonparametric AUC											
			Distribution of 1 Chi-Squared Tests × 1000 Experiments								
Pnts	Freq	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
16	1/8	32	0	0	0	0	0	1	0	0	0	0
16	1/8	64	0	0	1	0	0	0	0	0	0	0
16	1/8	128	0	0	1	0	0	0	0	0	0	0

Table B.5 ...end

Parametric ROC Parameters for 2 Points										
	Distribution of 50 Chi-Squared Tests × 2000 Experiments									
Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
1	0	1	4	14	16	8	6	1	0	0
2	0	1	6	10	11	12	6	1	3	0
4	0	0	2	13	14	12	7	2	0	0
8	0	0	4	10	16	9	6	4	0	1
16	0	0	4	8	13	15	8	0	2	0
32	0	2	6	18	13	7	4	0	0	0
64	0	1	5	10	12	13	5	1	3	0
128	0	1	4	10	15	4	9	4	3	0

Table B.6 Parametric ROC Parameters for 2 Points

Parametric ROC AUC for 2 Points										
	Distribution of 50 Chi-Squared Tests × 2000 Experiments									
Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
1	0	0	11	20	8	6	3	2	0	0
2	0	3	11	13	14	7	1	1	0	0
4	0	2	14	18	12	3	1	0	0	0
8	0	2	14	18	9	5	2	0	0	0
16	0	2	11	13	8	12	2	1	1	0
32	0	2	14	16	13	3	2	0	0	0
64	0	2	8	16	10	6	6	1	1	0
128	0	1	10	13	11	10	4	1	0	0

Table B.7 Parametric ROC AUC for 2 Points

Parametric ROC Parameters for 3 Points with 0.50 pdf											
		Distribution of 50 Chi-Squared Tests									
Exp	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
2000	1	0	2	13	16	10	1	5	3	0	0
2000	2	0	3	7	15	11	7	2	5	0	0
2000	4	0	3	13	11	14	6	1	1	1	0
2000	8	0	2	10	13	15	7	2	1	0	0
5000	16	0	1	6	12	18	10	2	1	0	0
10000	32	0	1	1	9	8	16	9	3	2	1
20000	64	0	0	1	5	4	8	11	6	7	8
20000	128	0	0	1	1	6	8	15	5	6	8

Table B.8 Parametric ROC Parameters for 3 Points with 0.50 pdf

Parametric ROC Parameters for 3 Points with 0.80 pdf											
		Distribution of 50 Chi-Squared Tests									
Exp	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
2000	1	0	5	12	17	13	2	0	1	0	0
2000	2	0	2	10	22	12	2	2	0	0	0
2000	4	0	4	11	18	10	4	3	0	0	0
2000	8	0	1	5	17	15	9	3	0	0	0
5000	16	0	2	10	20	12	3	3	0	0	0
10000	32	0	3	10	14	14	7	2	0	0	0
20000	64	0	1	3	13	12	10	8	3	0	0
20000	128	0	0	9	14	16	7	1	2	1	0

Table B.9 Parametric ROC Parameters for 3 Points with 0.80 pdf

Parametric ROC Parameters for 3 Points with 1.00 pdf											
		Distribution of 50 Chi-Squared Tests									
Exp	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
2000	1	0	2	10	15	13	5	5	0	0	0
2000	2	0	3	7	20	13	5	2	0	0	0
2000	4	0	2	12	18	8	6	2	1	0	1
2000	8	0	2	12	14	18	3	1	0	0	0
5000	16	0	1	10	16	9	13	1	0	0	0
10000	32	0	1	18	14	6	9	2	0	0	0
20000	64	0	1	15	13	13	5	1	1	1	0
20000	128	0	6	14	14	11	5	0	0	0	0

Table B.10 Parametric ROC Parameters for 3 Points with 1.00 pdf

Parametric ROC Parameters for 3 Points with 1.25 pdf											
		Distribution of 50 Chi-Squared Tests									
Exp	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
2000	1	0	4	17	13	10	5	1	0	0	0
2000	2	0	1	11	16	12	6	2	2	0	0
2000	4	0	2	11	12	12	7	4	1	1	0
2000	8	0	4	10	15	13	5	2	1	0	0
5000	16	0	1	7	17	13	4	6	1	1	0
10000	32	0	1	7	19	14	7	1	1	0	0
20000	64	0	2	9	13	11	9	3	2	1	0
20000	128	0	1	3	19	17	6	2	0	2	0

Table B.11 Parametric ROC Parameters for 3 Points with 1.25 pdf

Parametric ROC Parameters for 3 Points with 2.00 pdf											
		Distribution of 50 Chi-Squared Tests									
Exp	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
2000	1	0	0	4	10	10	8	11	5	1	1
2000	2	0	0	2	7	12	9	13	6	1	0
2000	4	0	0	2	5	13	14	12	1	3	0
2000	8	0	0	2	7	14	15	6	4	0	2
5000	16	0	0	1	0	5	10	13	11	5	5
10000	32	0	0	0	0	1	1	7	5	13	23
20000	64	0	0	0	0	0	0	0	2	4	44
20000	128	0	0	0	0	0	0	0	4	8	38

Table B.12 Parametric ROC Parameters for 3 Points with 2.00 pdf

Parametric ROC AUC for 3 Points with 0.50 pdf											
		Distribution of 50 Chi-Squared Tests									
Exp	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
2000	1	0	0	15	17	15	1	2	0	0	0
2000	2	1	0	5	16	15	8	1	1	2	1
2000	4	0	2	8	13	17	4	5	0	0	1
2000	8	0	2	12	8	15	4	5	4	0	0
5000	16	0	0	6	13	10	13	6	1	1	0
10000	32	0	0	0	13	8	10	9	5	3	2
20000	64	0	0	1	2	10	8	10	9	3	7
20000	128	0	0	1	1	8	11	8	8	6	7

Table B.13 Parametric ROC AUC for 3 Points with 0.50 pdf

Parametric ROC AUC for 3 Points with 0.80 pdf											
		Distribution of 50 Chi-Squared Tests									
Exp	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
2000	1	0	3	11	15	14	4	2	0	1	0
2000	2	0	0	10	19	14	3	4	0	0	0
2000	4	0	2	9	14	17	6	1	0	1	0
2000	8	0	3	13	14	14	5	1	0	0	0
5000	16	0	2	10	21	11	5	1	0	0	0
10000	32	0	0	7	13	17	8	2	2	1	0
20000	64	0	0	3	19	13	10	5	0	0	0
20000	128	0	2	8	12	14	8	4	0	1	1

Table B.14 Parametric ROC AUC for 3 Points with 0.80 pdf

Parametric ROC AUC for 3 Points with 1.00 pdf											
		Distribution of 50 Chi-Squared Tests									
Exp	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
2000	1	0	1	14	16	12	7	0	0	0	0
2000	2	0	2	8	21	10	8	1	0	0	0
2000	4	0	1	15	14	11	8	1	0	0	0
2000	8	0	1	11	14	13	10	1	0	0	0
5000	16	0	4	12	15	6	7	5	1	0	0
10000	32	0	3	10	14	13	6	4	0	0	0
20000	64	0	2	12	17	12	5	0	2	0	0
20000	128	0	7	7	12	12	8	4	0	0	0

Table B.15 Parametric ROC AUC for 3 Points with 1.00 pdf

Parametric ROC AUC for 3 Points with 1.25 pdf											
		Distribution of 50 Chi-Squared Tests									
Exp	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
2000	1	0	2	13	18	9	6	2	0	0	0
2000	2	0	1	6	19	15	6	2	1	0	0
2000	4	0	1	12	15	13	8	1	0	0	0
2000	8	1	0	8	18	11	7	4	1	0	0
5000	16	0	1	11	15	17	3	2	1	0	0
10000	32	0	3	4	16	12	10	5	0	0	0
20000	64	0	2	6	14	16	6	4	1	1	0
20000	128	0	1	4	14	16	8	4	3	0	0

Table B.16 Parametric ROC AUC for 3 Points with 1.25 pdf

Parametric ROC AUC for 3 Points with 2.00 pdf											
		Distribution of 50 Chi-Squared Tests									
Exp	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
2000	1	0	0	5	5	13	10	7	6	3	1
2000	2	0	0	2	4	12	13	13	5	1	0
2000	4	0	0	1	5	13	13	10	5	2	1
2000	8	0	0	1	4	13	19	4	6	2	1
5000	16	0	0	0	3	4	8	11	12	6	6
10000	32	0	0	0	0	0	5	4	5	10	26
20000	64	0	0	0	0	0	0	0	3	2	45
20000	128	0	0	0	0	1	1	2	7	6	33

Table B.17 Parametric ROC AUC for 3 Points with 2.00 pdf

Parametric ROC Parameters for 4 Points with 0.50 pdf											
		Distribution of 50 Chi-Squared Tests									
Exp	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
3000	1	0	1	9	18	13	5	1	2	1	0
4000	2	0	4	7	25	5	8	1	0	0	0
5000	4	0	1	11	18	11	7	1	1	0	0
10000	8	0	5	10	19	12	2	1	1	0	0
20000	16	0	2	11	14	14	9	0	0	0	0
20000	32	0	8	10	14	12	4	2	0	0	0
20000	64	0	0	14	20	10	3	2	1	0	0

Table B.18 Parametric ROC Parameters for 4 Points with 0.50 pdf

Parametric ROC Parameters for 4 Points with 0.80 pdf											
		Distribution of 50 Chi-Squared Tests									
Exp	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
3000	1	0	1	14	17	11	4	2	1	0	0
4000	2	0	4	8	15	13	6	3	1	0	0
5000	4	0	5	11	12	12	6	2	1	1	0
10000	8	0	2	16	19	10	2	1	0	0	0
20000	16	0	0	13	18	12	5	1	1	0	0
20000	32	0	0	7	17	16	6	3	0	0	1
20000	64	0	2	13	11	10	11	2	0	1	0

Table B.19 Parametric ROC Parameters for 4 Points with 0.80 pdf

Parametric ROC Parameters for 4 Points with 1.00 pdf											
		Distribution of 50 Chi-Squared Tests									
Exp	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
3000	1	0	2	19	10	10	7	2	0	0	0
4000	2	0	2	14	21	10	2	1	0	0	0
5000	4	0	4	14	18	7	6	1	0	0	0
10000	8	0	1	15	17	7	5	4	0	1	0
20000	16	0	9	7	19	12	2	0	1	0	0
20000	32	0	4	10	12	14	3	4	2	1	0
20000	64	0	1	14	13	15	4	2	0	1	0

Table B.20 Parametric ROC Parameters for 4 Points with 1.00 pdf

Parametric ROC Parameters for 4 Points with 1.25 pdf											
		Distribution of 50 Chi-Squared Tests									
Exp	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
3000	1	0	1	11	17	12	5	4	0	0	0
4000	2	1	0	12	13	12	8	4	0	0	0
5000	4	0	0	16	16	11	6	0	1	0	0
10000	8	0	3	15	14	8	8	2	0	0	0
20000	16	0	4	13	16	10	5	1	0	1	0
20000	32	0	5	17	8	12	6	1	1	0	0
20000	64	0	6	6	14	14	6	2	2	0	0

Table B.21 Parametric ROC Parameters for 3 Points with 1.25 pdf

Parametric ROC Parameters for 4 Points with 2.00 pdf											
		Distribution of 50 Chi-Squared Tests									
Exp	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
3000	1	0	2	9	17	12	8	2	0	0	0
4000	2	0	2	13	13	13	7	2	0	0	0
5000	4	0	2	9	18	11	6	2	2	0	0
10000	8	0	0	7	18	17	6	1	0	1	0
20000	16	0	3	12	15	11	8	0	1	0	0
20000	32	0	1	14	15	16	3	1	0	0	0
20000	64	0	1	18	12	10	6	3	0	0	0

Table B.22 Parametric ROC Parameters for 4 Points with 2.00 pdf

Parametric ROC AUC for 4 Points with 0.50 pdf											
		Distribution of 50 Chi-Squared Tests									
Exp	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
3000	1	0	5	8	18	10	9	0	0	0	0
4000	2	0	3	12	17	10	7	0	1	0	0
5000	4	0	3	11	20	12	3	1	0	0	0
10000	8	0	0	11	21	10	6	1	1	0	0
20000	16	0	2	14	14	10	4	5	1	0	0
20000	32	0	0	9	27	4	5	3	1	1	0
20000	64	0	3	14	11	12	7	1	2	0	0

Table B.23 Parametric ROC AUC for 3 Points with 0.50 pdf

Parametric ROC AUC for 4 Points with 0.80 pdf											
		Distribution of 50 Chi-Squared Tests									
Exp	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
3000	1	0	4	8	20	13	4	1	0	0	0
4000	2	0	3	13	15	12	6	1	0	0	0
5000	4	0	2	17	12	12	4	2	0	0	1
10000	8	0	3	9	20	9	4	4	1	0	0
20000	16	1	1	7	18	15	6	1	1	0	0
20000	32	0	1	11	18	13	4	3	0	0	0
20000	64	0	5	15	15	12	2	1	0	0	0

Table B.24 Parametric ROC AUC for 4 Points with 0.80 pdf

Parametric ROC AUC for 4 Points with 1.00 pdf											
		Distribution of 50 Chi-Squared Tests									
Exp	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
3000	1	0	2	15	17	7	7	1	0	0	1
4000	2	0	1	12	20	7	6	4	0	0	0
5000	4	0	2	13	20	11	2	1	1	0	0
10000	8	0	3	11	15	13	3	4	1	0	0
20000	16	0	3	11	15	14	5	2	0	0	0
20000	32	0	1	12	15	18	3	1	0	0	0
20000	64	0	3	9	18	11	8	1	0	0	0

Table B.25 Parametric ROC AUC for 4 Points with 1.00 pdf

Parametric ROC AUC for 4 Points with 1.25 pdf											
		Distribution of 50 Chi-Squared Tests									
Exp	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
3000	1	0	2	9	18	12	5	3	1	0	0
4000	2	0	8	11	15	9	7	0	0	0	0
5000	4	0	2	16	18	9	3	2	0	0	0
10000	8	0	3	9	13	12	7	6	0	0	0
20000	16	0	2	15	17	9	7	0	0	0	0
20000	32	0	3	9	15	15	4	3	1	0	0
20000	64	0	4	6	20	13	5	1	0	1	0

Table B.26 Parametric ROC AUC for 4 Points with 1.25 pdf

Parametric ROC AUC for 4 Points with 2.00 pdf											
		Distribution of 50 Chi–Squared Tests									
Exp	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
3000	1	0	1	10	18	15	5	0	1	0	0
4000	2	0	3	14	14	15	1	3	0	0	0
5000	4	0	3	12	12	13	6	3	0	1	0
10000	8	0	3	9	11	15	11	1	0	0	0
20000	16	0	2	10	13	17	2	3	3	0	0
20000	32	0	3	12	12	11	8	4	0	0	0
20000	64	0	2	7	24	10	6	0	1	0	0

Table B.27 Parametric ROC AUC for 4 Points with 2.00 pdf

Weighted Confusion Matrix											
		Distribution of 40 Chi–Squared Tests × 2000 Experiments									
Wgts	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
2	1	0	3	6	12	10	3	6	0	0	0
2	2	0	1	4	11	17	3	2	1	1	0
2	4	0	3	8	14	9	3	3	0	0	0
2	8	0	0	7	14	8	4	6	1	0	0
2	16	0	1	6	11	10	5	5	2	0	0
2	32	0	0	4	14	10	9	2	0	0	1
2	64	0	4	8	11	10	5	1	1	0	0
2	128	1	1	5	14	8	7	2	1	0	1
3	1	0	2	8	13	10	5	2	0	0	0
3	2	0	1	9	15	9	5	1	0	0	0
3	4	0	1	4	20	6	6	2	0	1	0
3	8	0	5	6	13	11	4	1	0	0	0
3	16	0	3	7	14	9	6	1	0	0	0
3	32	0	1	12	12	8	6	1	0	0	0
3	64	0	1	7	15	10	4	2	1	0	0
3	128	0	0	5	11	11	11	2	0	0	0
5	1	0	3	5	14	11	3	4	0	0	0
5	2	0	1	13	10	9	4	2	1	0	0
5	4	0	1	6	17	10	3	1	2	0	0
5	8	0	2	10	9	13	5	1	0	0	0
5	16	0	1	7	13	13	3	1	1	1	0
5	32	0	3	9	12	11	3	2	0	0	0
5	64	0	1	9	17	6	4	3	0	0	0
5	128	0	4	6	11	10	7	1	1	0	0
9	1	0	1	10	10	12	4	2	1	0	0
9	2	0	1	8	12	14	4	1	0	0	0
9	4	0	1	7	12	8	9	3	0	0	0
9	8	0	2	6	20	9	2	0	1	0	0
9	16	0	6	2	15	9	6	1	0	1	0
9	32	0	1	10	12	9	6	0	2	0	0
9	64	0	3	9	16	7	3	0	1	1	0
9	128	0	3	8	8	14	3	4	0	0	0
17	1	0	0	2	15	13	5	1	4	0	0

Table B.28 Weighted Confusion Matrix

Weighted Confusion Matrix											
		Distribution of 40 Chi–Squared Tests × 2000 Experiments									
Wgts	Case	≥ 0 < 5	≥ 5 <10	≥10 <15	≥15 <20	≥20 <25	≥25 <30	≥30 <35	≥35 <40	≥40 <45	≥45
17	2	0	0	1	11	13	9	4	2	0	0
17	4	0	2	5	11	12	7	3	0	0	0
17	8	0	1	5	13	14	4	2	1	0	0
17	16	0	2	13	10	10	5	0	0	0	0
17	32	0	0	7	15	9	6	2	1	0	0
17	64	1	2	13	8	7	4	2	3	0	0
17	128	0	1	11	11	9	6	2	0	0	0

Table B.28 ... end

C Software

This appendix lists the complete C++ source code for five minimal example programs to calculate the pdfs for all the novel methods discussed in Chapter 5:

- Nonparametric ROC point pdf – section C.1
- Nonparametric ROC point comparison pdf – section C.2
- Nonparametric AUC pdf – section C.3
- Parametric AUC pdf and parameters pdf – section C.4
- Weighted confusion matrix weight pdf – section C.5

For brevity and clarity, each example program is designed to show only the absolute essentials of initialising the look-up tables and calling the routines that calculate the pdfs. Example data is hard coded into the programs, and the generated pdfs are unused.

The five main programs and the libraries (files of functions) they use are listed below. Each library is listed below the first program to use it. To ease documentation, the libraries are linked to the main programs by the C++ preprocessor 'include' directive. Each include directive is followed by a reference to the section number where the library listing can be found. Libraries are subdivided into the functions they contain.

The REAL floating point class (section 5.2.7.1) has been omitted for brevity so the example programs are restricted to double precision calculations.

Copies of all the executable programs used to produce the graphs in Chapter 7 are provided on a 3.5" floppy disk with this thesis. These programs run from the command line (DOS box) of a PC running one of the Microsoft operating systems (Windows 3.x, NT etc.). A plain ASCII text file, 'ReadMe.txt', is provided on the disk giving further information, and each program will print out instructions if run without any command line arguments. Example data files have also been provided to assist in using the software. Some of the grid sizes in the example programs have been reduced from those used to produce the graphs in Chapter 7 so that the programs run in reasonable time and require reasonable amounts of memory.

C.1 Program for Nonparametric ROC Points

```
//                          DemoNonPara.cpp
//                          ===============
//
#include "DemoReal.cpp"     // C.1.1
#include "InitPower.cpp"    // C.1.2
#include "MakeSurface.cpp"  // C.1.3
#include "NonParaPdf.cpp"   // C.1.4

    const  long    One              =  512; // Grid size (One * One)
    const  long    MaxPower         = 1024; // Maximum power term

    int main
//  ========
    (
    )
    {

//      Initialise look-up tables

        double ** Power;
        double *  Factorial;

        InitAntiLog ( );
        InitPower ( Power, One, MaxPower );
        InitFactorial ( Factorial, MaxPower );

//      Set up the data for the ROC point

//      If using more than one ROC point the number of degrees of freedom need
//      to be accounted for

        long TruePos  = 8;    long FalsePos =  6;
        long FalseNeg = 2;    long TrueNeg  = 14;

//      Calculate the pdf of the Hit Rate and False Alarm Rate

        double FalseAlarmRatePdf [One];
        double HitRatePdf        [One];
        NonParaPdf (FalseAlarmRatePdf, One, Power, Factorial, FalsePos, TrueNeg );
        NonParaPdf (HitRatePdf,        One, Power, Factorial, TruePos,  FalseNeg);

//      Multiply pdfs together to get the 2D pdf of the ROC point

        double Surface[One*One];
        MakeSurface ( Surface, FalseAlarmRatePdf, HitRatePdf, One );

//      Tidy up

        FreeFactorial ( Factorial );
        FreePower ( Power, One );

        return 0;
    }
```

C.1.1 DemoReal File

```
//                          DemoReal.cpp
//                          ============
//
    const   long ZeroAntiLog =      1024;
    const   long MinAntiLog  = -ZeroAntiLog;
    const   long MaxAntiLog  =  ZeroAntiLog;
    const   long LenAntiLog  =  MaxAntiLog - MinAntiLog + 1;

    double  AntiLog[LenAntiLog];
```

C.1.1.1 InitAntiLog

```
      void InitAntiLog
//    ================
      (
      )
      {
        double Fraction = 1.0;
        for ( long A=0; A>=MinAntiLog; A-- ) {
          AntiLog[ZeroAntiLog+A] = Fraction;
          Fraction *= 0.5;
        }
        Fraction = 1.0;
        for ( A=0; A<=MaxAntiLog; A++ ) {
          AntiLog[ZeroAntiLog+A] = Fraction;
          Fraction *= 2.0;
        }
      }
```

C.1.1.2 AntiLog

```
      inline double Antilog
//    =====================
      (
        long A
      )
      {
        return AntiLog[ZeroAntiLog+A];
      }
```

C.1.1.3 Double

```
      inline double Double
//    ====================
      (
        double B
      )
      {
//      REAL version of this overloaded function converts a REAL to double

        return B;
      }
```

C.1.1.4 Norm

```
      inline double Norm
//    ==================
      (
        double A
      )
      {
//      REAL version of this overloaded function normalises the exponent
//      Thus is necessary because other REAL arithmetic operators do not normalise
//      in order to give extra speed

        return A;
      }
```

C.1.2 InitPower File

```
//
//                          InitPower.cpp
//                          =============
//
      #include <stdio.h>
      #include <stdlib.h>
```

C.1.2.1 InitPower

```
      template <class NUM> void InitPower
//    ===================================
```

```
    (
      NUM     ** &Power,
      long       One,
      long       MaxPower
    )
    {
      long MaxProb = One+1;

      if ( NULL == ( Power = (NUM**) malloc ( sizeof(NUM*) * MaxProb ) ) ) {
        printf ( "Can't allocate memory for 'Power'\n" );
        exit(1);
      }

      for ( long Prob=0; Prob<MaxProb; Prob++ ) {
        if ( NULL == ( Power[Prob] = (NUM*) malloc ( sizeof(NUM) * MaxPower)) ) {
          printf ( "Can't allocate memory for `Power'\n" );
          exit(1);
        }
      }
//                              Power
//                 0       1       2       3       4
//               +------+------+------+------+------+
//        0.0000|1.0000|0.0000|0.0000|0.0000|0.0000|
//               +------+------+------+------+------+
//        0.2500|1.0000|0.2500|0.0625|0.0156|0.0039|
//               +------+------+------+------+------+
// Prob  0.5000|1.0000|0.5000|0.2500|0.1250|0.0625|
//               +------+------+------+------+------+
//        0.7500|1.0000|0.7500|0.5625|0.4219|0.3164|
//               +------+------+------+------+------+
//        1.0000|1.0000|1.0000|1.0000|1.0000|1.0000|
//               +------+------+------+------+------+
//
//     Fill in the cumulative probabilities of Hits number of Prob probability

      for ( Prob=1; Prob<MaxProb; Prob++ ) {

        Power[Prob][0] = 1.0;
        Power[Prob][1] = Norm ( Prob / (double) One );

        for ( long Hits=2; Hits<MaxPower; Hits++ ) {
          Power[Prob][Hits] = Norm ( Power[Prob][Hits-1] * Power[Prob][1] );
        }
      }

      Power[0][0] = 1.0;

      for ( long Hits=1; Hits<MaxPower; Hits++ ) {
        Power[0][Hits] = 0.0;
      }
    }
```

C.1.2.2 FreePower

```
    template <class NUM> void FreePower
//  ===================================
    (
      NUM     ** &Power,
      long       One
    )
    {
      long MaxProb = One+1;

      for ( long Prob=0; Prob<MaxProb; Prob++ ) {
        free ( Power[Prob] );
      }

      free ( Power );

    }
```

C.1.3 MakeSurface File

```
//                              MakeSurface.cpp
//                              ===============
//
```

C.1.3.1 MakeSurface

```
      void MakeSurface
//    ================
      (
        double * Surface,
        double * PdfX,
        double * PdfY,
        long     DimSur
      )
      {

        for ( long X=0; X<DimSur; X++ ) {
          for ( long Y=0; Y<DimSur; Y++ ) {
            Surface[X*DimSur+Y] = PdfX[X] * PdfY[Y];
          }
        }
      }
```

C.1.4 NonParaPdf File

```
//                              NonParaPdf.cpp
//                              ==============
//
      #include <math.h>
```

C.1.4.1 InitFactorial

```
      template <class NUM> void InitFactorial
//    =======================================
      (
        NUM   * & Factorial,
        long      MaxPower
      )
      {

        if ( NULL == ( Factorial = (NUM*) malloc ( sizeof(NUM) * MaxPower ) ) ) {
          printf ( "Can't allocate memory for 'Factorial'\n" );
          exit(1);
        }

        Factorial[0] = 1.0;

        for ( long N=1; N<MaxPower; N++ ) {
          Factorial[N] = Norm ( Factorial[N-1] * N );
        }
      }
```

C.1.4.2 FreeFactorial

```
      template <class NUM> void FreeFactorial
//    =======================================
      (
        NUM   * & Factorial
      )
      {
        free ( Factorial );
      }
```

C.1.4.3 NonParaPdf

```
      template <class NUM> NonParaPdf
//    ===============================
```

```
(
  double * Pdf,
  long     N,
  NUM **   Power,
  NUM *    Factorial,
  long     A0,
  long     A1
)
{
  NUM Recriprocal [MaxPower];

  long AA1 = A0 + A1 + 1;

  long Smallest;
  if ( A1 < A0 ) {
    Smallest = A1;
  } else {
    Smallest = A0;
  }

//   Pre-calculate for speed (given a lookup table of factorials)

  for ( long K=0; K<=Smallest; K++ ) {
    Recriprocal[K] =
      Factorial[AA1] / ( Factorial[K] * Factorial[AA1-K] );
  }

//   Take advantage of equation symmetry for speed

  if ( A1 < A0 ) {

//     Calculate pdf from look-up table of powers

    double Last = 0.0;
    for ( long I=1; I<=N; I++ ) {
      double BoundaryVal = 0.0;
      for ( long K=0; K<=A1; K++ ) {
        BoundaryVal +=
          Double ( Power[I][AA1-K] * Power[N-I][K] * Recriprocal[K] );

//          ( Double(x) converts a value of type x to type double )

      }
      Pdf[I-1] = BoundaryVal - Last;
      Last = BoundaryVal;
    }

  } else {

    double Last = 1.0;
    for ( long I=1; I<=N; I++ ) {
      double  BoundaryVal = 0.0;
      for ( long K=0; K<=A0; K++ ) {
        BoundaryVal +=
          Double ( Power[N-I][AA1-K] * Power[I][K] * Recriprocal[K] );
      }
      Pdf[I-1] = Last - BoundaryVal;
      Last = BoundaryVal;
    }

  }
}
```

C.2 Program for Nonparametric Comparison

```
//                    DemoNonParaCmp.cpp
//                    ==================
//
#include "DemoReal.cpp" // C.1.1
```

```
#include "InitPower.cpp" // C.1.2
#include "NonParaPdf.cpp" // C.1.4
#include "NonParaCmpPdf.cpp" // C.2.1
#include "MakeSurface.cpp"  // C.1.3

    const  long    One             = 512;  // ROC grid size
    const  long    MaxCmp          = One+One-1; // Comparison
                                                // grid size
    const  long    MaxPower        = 1024; // Maximum power term

    const  long    Exp0            =   0;  // Experiment no.
    const  long    Exp1            =   1;
    const  long    MaxExp          =   2;

    void main
//  =========
    (
    )
    {

//    Initialise look-up tables

      double ** Power;
      double *  Factorial;
      InitAntiLog ( );
      InitPower ( Power, One, MaxPower );
      InitFactorial ( Factorial, MaxPower );

//    Set up the data for two ROC points

      long TruePos[MaxExp];  long FalsePos[MaxExp];
      long FalseNeg[MaxExp]; long TrueNeg[MaxExp];

      TruePos[Exp0]  = 8;    FalsePos[Exp0] =  6;
      FalseNeg[Exp0] = 2;    TrueNeg[Exp0]  = 14;

      TruePos[Exp1]  = 9;    FalsePos[Exp1] =  6;
      FalseNeg[Exp1] = 1;    TrueNeg[Exp1]  = 14;

//    Calculate the pdf of both Hit Rates and False Alarm Rates

      double FalseAlarmRatePdf[MaxExp][One];
      double HitRatePdf[MaxExp][One];
      for ( long Exp=0; Exp<MaxExp; Exp++ ) {

       NonParaPdf ( FalseAlarmRatePdf[Exp], One, Power, Factorial,
         FalsePos[Exp], TrueNeg[Exp] );
       NonParaPdf ( HitRatePdf[Exp],        One, Power, Factorial,
         TruePos[Exp],  FalseNeg[Exp]);

      }

//    Compare the pdfs

      double FalseAlarmRateCmpPdf[MaxCmp];
      double HitRateCmpPdf[MaxCmp];
      NonParaCmpPdf ( FalseAlarmRateCmpPdf, FalseAlarmRatePdf[Exp0],
        FalseAlarmRatePdf[Exp1], One );
      NonParaCmpPdf ( HitRateCmpPdf,        HitRatePdf[Exp0],
        HitRatePdf[Exp1],        One );

//    Multiply comparison pdfs together to get the 2D pdf of the difference

      double DiffSurface[MaxCmp*MaxCmp];
      MakeSurface ( DiffSurface, FalseAlarmRateCmpPdf, HitRateCmpPdf, MaxCmp );

//    Tidy up

      FreeFactorial ( Factorial );
      FreePower ( Power, One );
```

```
}
```

C.2.1 NonParaCmpPdf File

```
//                          NonParaCmpPdf.cpp
//                          =================
//
```

C.2.1.1 NonParaCmpPdf

```
      void NonParaCmpPdf
//    ==================
      (
        double * CmpPdf,
        double * Pdf0,
        double * Pdf1,
        long     One
      )
      {

        long MaxCmp = One+One-1;
        for ( long X=0; X<MaxCmp; X++ ) {
          CmpPdf[X] = 0.0;
        }

        for ( long X0=0; X0<One; X0++ ) {
          for ( long X1=0; X1<One; X1++ ) {
            CmpPdf[One-1+X0-X1] += Pdf0[X0] * Pdf1[X1];
          }
        }

      }
```

C.3 Program for Nonparametric AUC

```
//                          DemoNonParaAuc.cpp
//                          =================
//
#include "DemoReal.cpp" // C.1.1
#include "InitPower.cpp" // C.1.2

      const  long     One              = 128;
      const  long     MaxProb          = One+1;

      const  long     MaxPts           = 16;
      const  long     MaxCat           = MaxPts+1;

      const  long     MaxPower         = 1024;

      const  long     Dis              =   0;
      const  long     Hlt              =   1;
      const  long     MaxGold          =   2;

#include "NonParaAucPdf.cpp" // C.3.1

      void main
//    =========
      (
      )
      {

//      Initialise look-up tables

        double ** Power;
        InitAntiLog ( );
        InitPower ( Power, One, MaxPower );
```

```
    InitArea ( );

    long Cases [MaxGold][MaxCat];

//                      Gold Standard
//          Diseased              Healthy
//                                                    Test
    Cases[Dis][0] =  8;  Cases[Hlt][0] =  2; // Diseased
    Cases[Dis][1] =  3;  Cases[Hlt][1] =  5; // Unknown
    Cases[Dis][2] =  1;  Cases[Hlt][2] = 12; // Healthy

    long NoPts = 2; // Number of ROC points (No.Categories-1)

    double AUCPdf[MaxProb];
    NonParaAucPdf ( AUCPdf, Power, Cases, NoPts );

    FreePower ( Power, One );
}
```

C.3.1 NonParaAucPdf File

```
//                      NonParaAucPdf.cpp
//                      =================
//
          const    long        MrySpace    =    MaxProb*MaxProb*(MaxProb/4+2);
static long MaxArea[MaxProb][MaxProb];
```

C.3.1.1 InitArea

```
    void InitArea
//  =============
    (
    )
    {
      long Total = 0;
      for ( long Y=0; Y<MaxProb; Y++ ) {
        for ( long X=0; X<MaxProb; X++ ) {
          MaxArea[Y][X] = (Y*X+One/2)/One;
          Total += MaxArea[Y][X] + 1;
        }
      }
      if ( Total >= MrySpace ) {
        printf ( "Total area (%d) >= than space for arrays (%d)\n",
          Total, MaxProb*MaxProb*MaxProb/4 );
        exit ( 1 );
      }
    }
```

C.3.1.2 DeltaArea

```
    inline long DeltaArea
//  =====================
    (
      long Xd,
      long Yo,
      long Yd
    )
    {
      return ((Yo+Yo+Yd)*Xd + 1)/(2*One);
    }
```

C.3.1.3 CalcArea

```
    long CalcArea
//  =============
    (
      long PopPts[MaxGold][MaxCat],
      long NoPts
    )
```

```
{
  long Area = 0;
  long X = 0;
  long Y = 0;
  for ( long P=0; P<=NoPts; P++ ) {
    Area += DeltaArea ( PopPts[Hit][P]-X, Y, PopPts[Dis][P]-Y );
    X = PopPts[Hit][P];
    Y = PopPts[Dis][P];
  }
  return Area;
}
```

C.3.1.4 NonParaAucPdf

```
template <class NUM> void NonParaAucPdf
//  ========================================
(
  double  Pdf[MaxProb],
  NUM **  Power,
  long    Cases[MaxGold][MaxCat],
  long    NoPts
)
{
  static NUM InSpace[MrySpace];
  static NUM RsSpace[MrySpace];
  static NUM * Wk[2][MaxProb][MaxProb];

  const NUM Zero = 0.0;

  long In = 0;
  long Rs = 1;

//  Allocate area columns only as long as the maximum area
//  at that location

  long SpcIdx = 0;
  for ( long Y=0; Y<MaxProb; Y++ ) {
    for ( long X=0; X<MaxProb; X++ ) {
      Wk[In][Y][X] = &InSpace[SpcIdx];
      Wk[Rs][Y][X] = &RsSpace[SpcIdx];
      SpcIdx += MaxArea[Y][X]+1;
      for ( long A=0; A<=MaxArea[Y][X]; A++ ) {
        Wk[In][Y][X][A] = Zero;
      }
      long Ad = DeltaArea ( X, 0, Y );
      Wk[In][Y][X][Ad] =
        Power[Y][Cases[Dis][0]] *
        Power[X][Cases[Hit][0]];
    }
  }

  for ( long N=1; N<NoPts; N++ ) {

//    Initialise and pre-calculate for speed

    static NUM Pw[MaxProb][MaxProb];
    for ( long Y=0; Y<MaxProb; Y++ ) {
      for ( long X=0; X<MaxProb; X++ ) {
        for ( long A=0; A<=MaxArea[Y][X]; A++ ) {
          Wk[Rs][Y][X][A] = Zero;
        }
        Pw[Y][X] =
          Power[Y][Cases[Dis][N]] *
          Power[X][Cases[Hit][N]];
      }
    }

//    Sum area over paths

    for ( long Yo=0; Yo<MaxProb; Yo++ ) {
```

```
            for ( long Yd=0; Yd<MaxProb-Yo; Yd++ ) {
              for ( long Xd=0; Xd<MaxProb; Xd++ ) {
                long Ad = DeltaArea ( Xd, Yo, Yd );
                for ( long Xo=0; Xo<MaxProb-Xd; Xo++ ) {
                  for ( long A=0; A<=MaxArea[Yo][Xo]; A++ ) {
                    Wk[Rs][Yo+Yd][Xo+Xd][A+Ad] +=
                      Wk[In][Yo][Xo][A] * Pw[Yd][Xd];
                  }
                }
              }
            }
          }

//        Cycle work arrays

          long Swap = In;
          In = Rs;
          Rs = Swap;
        }

//      Last segment of paths

        NUM Gather[MaxProb];
        for ( long A=0; A<MaxProb; A++ ) {
          Gather[A] = Zero;
        }

        for ( Y=0; Y<MaxProb; Y++ ) {
          for ( long X=0; X<MaxProb; X++ ) {

//          Pre-calculate

            NUM Pw =
              Power[One-Y][Cases[Dis][NoPts]] *
              Power[One-X][Cases[Hit][NoPts]];

//          Gather up paths

            long Ad = DeltaArea ( One-X, Y, One-Y );
            for ( long A=0; A<=MaxArea[Y][X]; A++ ) {
              Gather[A+Ad] += Wk[In][Y][X][A] * Pw;
            }
          }
        }

//      Normalise

        NUM Sum = Zero;
        for ( A=0; A<MaxProb; A++ ) {
          Sum += Gather[A];
        }

        NUM Recip = 1.0 / Sum;
        for ( A=0; A<MaxProb; A++ ) {
          Pdf[A] = Double ( Gather[A] * Recip );
        }

    }
```

C.4 Program for Parameter ROC Curves

```
//                              DemoPara.cpp
//                              ============
//
#include "DemoReal.cpp" // C.1.1
#include "InitPower.cpp" // C.1.2
#include "MaxMin.cpp" // C.4.1

    const  long     CopyIt          = 128;
```

```
    const  long    One              = CopyIt;
    const  long    OneU             = CopyIt;
    const  long    OneS             = CopyIt;
    const  long    OneA             = CopyIt;

    const  long    NotParaU         = OneU+1;
    const  long    NotParaS         = 0;
    const  long    NotParaA         = OneA+1;

//  Sizes of pdfs arrays (not including nonpara)

    const  long    MaxProb          = One+1;
    const  long    MaxU             = OneU+1;
    const  long    MaxS             = OneS+1;
    const  long    MaxA             = OneA+1;

//  Array sizes (including nonpara);

    const  long    DimProb          = MaxProb;
    const  long    DimU             = MaxU+1;
    const  long    DimS             = MaxS;
    const  long    DimA             = MaxA+1;

    const  long    MaxNoPts         =   5;
    const  long    MaxCat           = MaxNoPts+1;

    const  long    MaxPower         = 1024;

    const  long    Dis              =   0;
    const  long    Hlt              =   1;
    const  long    MaxGold          =   2;

#include "ParaPdf.cpp" // C.4.2

    void main
//  =========
    (
    )
    {

      double ** Power;
      InitPower ( Power, One, MaxPower );
      InitAntiLog ( );
      InitStdDev ( );
      InitClose ( );
      InitAuc ( );
      InitLines ( );

      long   Cases [MaxGold][MaxCat];

//                    Gold Standard
//          Diseased              Healthy
//                                                    Test
      Cases[Dis][0] =  8;  Cases[Hlt][0] =  2; // Diseased
      Cases[Dis][1] =  3;  Cases[Hlt][1] =  5; // Unknown
      Cases[Dis][2] =  1;  Cases[Hlt][2] = 12; // Healthy

      long NoPts = 2; // Number of ROC points (No.Categories-1)

      double MSdPdf[DimU][DimS];
      ParaPdf ( MSdPdf, Power, Cases, NoPts );

      double AucPdf[DimA];
      GenParaAucPdf ( AucPdf, MSdPdf );

      FreeLines ( );
      FreePower ( Power, One );
    }
```

C.4.1 MaxMin File

```
//                                        MaxMin.cpp
//                                        ==========
```

C.4.1.1 Max

```
      template <class NUM> inline NUM Max
//    ===================================
      (
        NUM   X,
        NUM   Y
      )
      {
        return (X>Y?X:Y);
      }
```

C.4.1.2 Min

```
      template <class NUM> inline NUM Min
//    ===================================
      (
        NUM   X,
        NUM   Y
      )
      {
        return (X<Y?X:Y);
      }
```

C.4.2 ParaPdf File

```
//                                        ParaPdf.cpp
//                                        ==========
//
      #include <math.h>

#define ONE unsigned char

      struct LINE
//    -----------
      {
        ONE XBgn;
        ONE YBgn;
        ONE XEnd;
        ONE YEnd;
      };

      struct CLOSE
//    ------------
      {
        double Dist;
        long   U;
        long   S;
      };

      long    NoLns[DimU][DimS];
      LINE *  Lines[DimU][DimS];
      long    Area[DimU][MaxS];
      CLOSE   Close[MaxProb][MaxProb];

      double StdDev[MaxProb];

      const double Factor = 1.09861228866811;
```

C.4.2.1 Sigmoid

```
      inline double Sigmoid
//    =====================
      (
```

```
    double X
  )
  {
    return ( 1.0 / ( 1.0 + exp ( - X * Factor ) ) );
  }
```

C.4.2.2 AntiSig

```
  inline double AntiSig
//  =====================
  (
    double Y
  )
  {
    return ( - log ( 1.0 / Y - 1.0 ) ) / Factor;
  }
//
//  Normal Distribution:
//
//  f(x) = 1 / sd * sqrt(2*Pi) * exp ( - (x-mu)*(x-mu) / 2*sd*sd );
//
//  Mean E(x) = mu
//  Variance Var(x) = sd * sd;
//
```

C.4.2.3 NormDis

```
  inline double NormDis
//  =====================
  (
    double U,
    double S
  )
  {
    return exp ( - U*U/(2*S*S) );
  }
```

C.4.2.4 AntiNorm

```
  inline double AntiNorm
//  ======================
  (
    double S,
    double D
  )
  {
    return sqrt ( -2*S*S * log ( D ) );
  }
```

C.4.2.5 InitStdDev

```
  void InitStdDev
//  ===============
  (
  )
  {
    const long    NoQuanta = One*128;  // Fractions of 1.0;
    const long    MinSd    = -8 * NoQuanta;
    const long    MaxSd    =  8 * NoQuanta;

    double Sum = 0.0;
    for ( long Sd=MinSd; Sd<=MaxSd; Sd++ ) {
      double U = Sd / (double) NoQuanta;
      Sum += NormDis ( U, 1.0 );
    }

    double Recip = 1.0/Sum;

    Sum = 0.0;
    long Idx = 0;
```

```
    double Test = 0.0;
    for ( Sd=MinSd; Sd<=MaxSd; Sd++ ) {
      double U = Sd / (double) NoQuanta;
      Sum += NormDis ( U, 1.0 ) * Recip;
      if ( Sum >= Test ) {
        StdDev[Idx] = U;
        Idx++;
        Test = Idx / ((double) One);
      }
    }
    StdDev[0]   = - AntiNorm ( 1.0, 1.0e-2 / (double) One );
    StdDev[One] = + AntiNorm ( 1.0, 1.0e-2 / (double) One );

  }
```

C.4.2.6 InitClose

```
  void InitClose
//============
  (
  )
  {
    for ( long X=0; X<MaxProb; X++ ) {
      for ( long Y=0; Y<MaxProb; Y++ ) {
        Close[X][Y].Dist = 1.0;
        Close[X][Y].U    = NotParaU;
        Close[X][Y].S    = NotParaS;
      }
    }
  }
```

C.4.2.7 AreaAtPnt

```
  inline double AreaAtPnt
//======================
  (
    double      Du,
    double      Sh,
    double      Sd,
    long &      P,
    double *    PathX,
    double *    PathY,
    const long MaxPath,
    const long NoQuanta
  )
  {

    P = 0;

    double Rh = AntiNorm ( Sh, 1.0e-9 );
    double Rd = AntiNorm ( Sd, 1.0e-9 );

    double XSum = 0.0;
    double YSum = 0.0;
    double ASum = 0.0;

//  Iterate in steps of the largest standard deviation.
//
//  Make sure an iteration step hits the mean of the smallest
//  deviation bang on.
//
//  E.g. consider a healthy sd of 2.0, and a disease sd of 0.0.
//  Step in 2.0/10 (0.2) rather than 0.0/10 (0.0)!
//  Calculate the cumulative at ... -0.2, 0.0, +2.0 ... of
//  the diseased distribution so that it jumps from 0.0 to 1.0
//  at 0.0, and then continues at 1.0! If the peak is missed
//  the cumulative stays at 0.0!

    if ( Sh < Sd ) {
      double Quanta = Sd / NoQuanta;
```

```
      long    MinSd = (long) ( Min ( -Rh+Du, -Rd ) / Quanta - 1 );
      long    MaxSd = (long) ( Max (  Rh+Du,  Rd ) / Quanta + 1 );

      for ( long T=MinSd; T<=MaxSd; T++ ) {
        double Uh = T * Quanta;

        double Ud = Uh + Du;
        double Xd = NormDis ( Uh, Sh );
        double Yd = NormDis ( Ud, Sd );

        ASum += (YSum+YSum+Yd)*Xd/2;
        XSum += Xd;
        YSum += Yd;

        PathX[P] = XSum;
        PathY[P] = YSum;
        P++;
        if ( P >= MaxPath ) {
          printf ( "MaxPath of %d too short\n", MaxPath );
          exit ( 1 );
        }
      }
    } else {
      double Quanta = Sh / NoQuanta;

      long    MinSd = (long) ( Min ( -Rh, -Rd+Du ) / Quanta - 1 );
      long    MaxSd = (long) ( Max (  Rh,  Rd+Du ) / Quanta + 1 );

      for ( long T=MinSd; T<=MaxSd; T++ ) {
        double Ud = T * Quanta;

        double Uh = Ud - Du;
        double Xd = NormDis ( Uh, Sh );
        double Yd = NormDis ( Ud, Sd );

        ASum += (YSum+YSum+Yd)*Xd/2;
        XSum += Xd;
        YSum += Yd;

        PathX[P] = XSum;
        PathY[P] = YSum;
        P++;
        if ( P >= MaxPath ) {
          printf ( "MaxPath of %d too short\n", MaxPath );
          exit ( 1 );
        }
      }
    }

    double RecipX = 1.0 / XSum;
    double RecipY = 1.0 / YSum;
    for ( long I=0; I<P; I++ ) {
      PathX[I] *= RecipX;
      PathY[I] *= RecipY;
    }

    return ASum / ( XSum*YSum );
  }
```

C.4.2.8 AreaAtPoint

```
  long AreaAtPoint
//  ================
  (
    long        U,
    long        S,
    long &      P,
    double *    PathX,
    double *    PathY,
```

```
    const long MaxPath,
    const long NoQuanta
  )
  {

    double Sh;
    if ( S == 0     ) {
      Sh = 2.0 / (double) ( OneS * 1.0e+6 );
    } else {
      Sh = 2 * S / (double) OneS;
    }

    double Sd;
    if ( S == OneS ) {
      Sd = 2.0 / (double) ( OneS * 1.0e+6 );
    } else {
      Sd = 2 * (OneS-S) / (double) OneS;
    }

    double Du;
    if ( U == 0 ) {
      Du = - AntiNorm ( 2.0, 1.0e-9 );
    } else if ( U == OneU ) {
      Du = + AntiNorm ( 2.0, 1.0e-9 );
    } else {
      Du = AntiSig ( U / (double) OneU );
    }

    double Area = AreaAtPnt ( Du, Sh, Sd, P,PathX,PathY,MaxPath,NoQuanta);

    for ( long I=0; I<P; I++ ) {
      long IntX = (long) ( PathX[I] * One + 0.5 );
      long IntY = (long) ( PathY[I] * One + 0.5 );
      double Dist = ( PathX[I]*One - IntX ) * ( PathX[I]*One - IntX ) +
                    ( PathY[I]*One - IntY ) * ( PathY[I]*One - IntY );
      if ( Dist < Close[IntX][IntY].Dist ) {
        Close[IntX][IntY].Dist = Dist;
        Close[IntX][IntY].U    = U;
        Close[IntX][IntY].S    = S;
      }
    }

    return (long) ( Area * OneA + 0.5 );
  }
```

C.4.2.9 InitAuc

```
  void InitAuc
//  ============
  (
  )
  {
    const long NoQuanta = 32;

    const long MaxPath = NoQuanta*40;
    static double PathX[MaxPath];
    static double PathY[MaxPath];

    long P;
    for ( long S=0; S<MaxS; S++ ) {
      Area[NotParaU][S] = NotParaA;
    }

    for ( long U=0; U<MaxU; U++ ) {
      for ( long S=0; S<MaxS; S++ ) {
        Area[U][S] = AreaAtPoint ( U, S, P, PathX, PathY, MaxPath, NoQuanta );
      }
    }
  }
```

C.4.2.10 Prob2Us

```
    inline void Prob2Us
//  ===================
    (
      double & U,
      double & Sh,
      long     XBgn,
      long     YBgn,
      long     XEnd,
      long     YEnd
    )
    {
      double dX = StdDev[XEnd] - StdDev[XBgn];
      double dY = StdDev[YEnd] - StdDev[YBgn];

      double Sum = dX + dY;

      Sh = 2 * dY / Sum;    // alternatively  Sd = 2 * dX / Sum

      U = (    (StdDev[YBgn]+StdDev[YEnd]) * dX
             - (StdDev[XBgn]+StdDev[XEnd]) * dY )
                          / Sum;
    }
```

C.4.2.11 ProbToUs

```
    void ProbToUs
//  =============
    (
      long & U,
      long & S,
      long   XBgn,
      long   YBgn,
      long   XEnd,
      long   YEnd
    )
    {
      if ( XBgn == XEnd && YBgn == YEnd ) {
        U = Close[XBgn][YBgn].U;
        S = Close[XBgn][YBgn].S;
      } else {

        double u, sd;
        Prob2Us ( u, sd, XBgn, YBgn, XEnd, YEnd );

        U = (long) ( OneU * Sigmoid ( u ) + 0.5 );
        S = (long) ( OneS * sd / 2.0 + 0.5 );

      }

    }
```

C.4.2.12 InitLines

```
    void InitLines
//  ==============
    (
    )
    {

      for ( long U=0; U<DimU; U++ ) {
        for ( long S=0; S<DimS; S++ ) {
          NoLns[U][S] = 0;
        }
      }

      for ( long X=0; X<MaxProb; X++ ) {
        for ( long Xd=0; Xd<MaxProb-X; Xd++ ) {
          for ( long Y=0; Y<MaxProb; Y++ ) {
```

```
            for ( long Yd=0; Yd<MaxProb-Y; Yd++ ) {
              long U;
              long S;
              ProbToUs ( U, S, X, Y, X+Xd, Y+Yd );
              NoLns[U][S]++;
            }
          }
        }
      }

      for ( U=0; U<DimU; U++ ) {
        for ( long S=0; S<DimS; S++ ) {
          if ( NoLns[U][S] > 0 ) {
            if ( NULL == ( Lines[U][S] =
                 (LINE*) malloc ( sizeof(LINE) * NoLns[U][S] ) ) ) {
              printf ( "Can't allocate memory for 'Lines'\n" );
              exit(1);
            }
          } else {
            Lines[U][S] = NULL;
          }
          NoLns[U][S] = 0;
        }
      }

      for ( X=0; X<MaxProb; X++ ) {
        for ( long Xd=0; Xd<MaxProb-X; Xd++ ) {
          for ( long Y=0; Y<MaxProb; Y++ ) {
            for ( long Yd=0; Yd<MaxProb-Y; Yd++ ) {

              long U;
              long S;
              ProbToUs ( U, S, X, Y, X+Xd, Y+Yd );

              Lines[U][S][NoLns[U][S]].XBgn = (ONE) X;
              Lines[U][S][NoLns[U][S]].YBgn = (ONE) Y;
              Lines[U][S][NoLns[U][S]].XEnd = (ONE) (X+Xd);
              Lines[U][S][NoLns[U][S]].YEnd = (ONE) (Y+Yd);

              NoLns[U][S]++;

            }
          }
        }
      }
    }
```

C.4.2.13 FreeLines

```
    void FreeLines
//  ==============
    (
    )
    {
      for ( long U=0; U<DimU; U++ ) {
        for ( long S=0; S<DimS; S++ ) {
          if ( NoLns[U][S] > 0 ) {
            free ( Lines[U][S] );
          }
        }
      }
    }
```

C.4.2.14 ParaPdf

```
    template <class NUM> void ParaPdf
//  =================================
    (
      double     Pdf[DimU][DimS],
      NUM     ** Power,
```

```
    long        Cases[MaxGold][MaxCat],
    long        NoPts
)
{
    long In = 0;
    long Rs = 1;
    static NUM Work[2][MaxProb][MaxProb];
    static NUM Mul[MaxCat][MaxProb][MaxProb];
    static NUM PdfTmp[DimU][DimS];
    const  NUM Zero = 0.0;

//  Pre-calculate for speed

    for ( long C=0; C<NoPts; C++ ) {
      for ( long X=0; X<MaxProb; X++ ) {
        for ( long Y=0; Y<MaxProb; Y++ ) {
          Mul[C][X][Y] =
            Power[X][Cases[Hlt][C]] *
            Power[Y][Cases[Dis][C]];
        }
      }
    }
    for ( long X=0; X<MaxProb; X++ ) {
      for ( long Y=0; Y<MaxProb; Y++ ) {
        Mul[NoPts][X][Y] =
          Power[One-X][Cases[Hlt][NoPts]] *
          Power[One-Y][Cases[Dis][NoPts]];
      }
    }

//  Calculate the nonparametric (including parametric) part

    for ( X=0; X<MaxProb; X++ ) {
      for ( long Y=0; Y<MaxProb; Y++ ) {
        Work[In][X][Y] = Mul[0][X][Y];
      }
    }

    for ( C=1; C<NoPts; C++ ) {
      for ( long X=0; X<MaxProb; X++ ) {
        for ( long Y=0; Y<MaxProb; Y++ ) {
          Work[Rs][X][Y] = Zero;
        }
      }
      for ( X=0; X<MaxProb; X++ ) {
        for ( long Xd=0; Xd<MaxProb-X; Xd++ ) {
          for ( long Y=0; Y<MaxProb; Y++ ) {
            for ( long Yd=0; Yd<MaxProb-Y; Yd++ ) {
              Work[Rs][X+Xd][Y+Yd] +=
                Work[In][X][Y] * Mul[C][Xd][Yd];
            }
          }
        }
      }
      long Swap = Rs;
      Rs = In;
      In = Swap;
    }

    NUM Total = Zero;
    for ( X=0; X<MaxProb; X++ ) {
      for ( long Y=0; Y<MaxProb; Y++ ) {
        Total += Work[In][X][Y] * Mul[NoPts][X][Y];
      }
    }

//  Calculate the parametric part by iterating over differences
//  in mean (U), and ratios of standard deviation (S)

    NUM Sum = Zero;
```

```
        for ( long U=0; U<MaxU; U++ ) {
          for ( long S=0; S<MaxS; S++ ) {

            for ( long X=0; X<MaxProb; X++ ) {
              for ( long Y=0; Y<MaxProb; Y++ ) {
                Work[In][X][Y] = Mul[0][X][Y];
              }
            }

            long Len = NoLns[U][S];

            for ( long C=1; C<NoPts; C++ ) {

              for ( long X=0; X<MaxProb; X++ ) {
                for ( long Y=0; Y<MaxProb; Y++ ) {
                  Work[Rs][X][Y] = Zero;
                }
              }

              for ( long I=0; I<Len; I++ ) {

                long Xb = (long) Lines[U][S][I].XBgn;
                long Yb = (long) Lines[U][S][I].YBgn;
                long Xe = (long) Lines[U][S][I].XEnd;
                long Ye = (long) Lines[U][S][I].YEnd;
                Work[Rs][Xe][Ye] += Work[In][Xb][Yb] * Mul[C][Xe-Xb][Ye-Yb];

              }

              long Swap = Rs;
              Rs = In;
              In = Swap;
            }

            PdfTmp[U][S] = 0.0;
            for ( X=0; X<MaxProb; X++ ) {
              for ( long Y=0; Y<MaxProb; Y++ ) {
                PdfTmp[U][S] += Work[In][X][Y] * Mul[NoPts][X][Y];
              }
            }
            Sum += PdfTmp[U][S];
          }
        }

//      Normalise

        NUM Recip = 1.0/Total;

        for ( U=0; U<MaxU; U++ ) {
          for ( long S=0; S<MaxS; S++ ) {
            Pdf[U][S] = Double ( PdfTmp[U][S] * Recip );
          }
        }

        for ( long S=0; S<MaxS; S++ ) {
          Pdf[NotParaU][S] = 0.0;
        }

        Pdf[NotParaU][NotParaS] = Double ( (Total-Sum) * Recip );

    }
```

C.4.2.15 GenParaAucPdf

```
    void GenParaAucPdf
//  ==================
    (
      double AucPdf[DimA],
      double Pdf[DimU][DimS]
    )
```

```
    {
      for ( long P=0; P<DimA; P++ ) {
        AucPdf[P] = 0.0;
      }
      for ( long U=0; U<MaxU; U++ ) {
        for ( long S=0; S<MaxS; S++ ) {
          AucPdf[Area[U][S]] += Pdf[U][S];
        }
      }
      AucPdf[NotParaA] = Pdf[NotParaU][NotParaS];

    }
```

C.5 Program for Weighted Confusion Matrix

```
//                              DemoWeight.cpp
//                              ==============
//
#include "DemoReal.cpp" // C.1.1
#include "InitPower.cpp" // C.1.2

    const  long     One             =  256;
    const  long     MaxProb         = One+1;

    const  long     MaxCat          =   16;
    const  long     MaxPower        = 1024;

    const  long     MaxWgt          = MaxProb;

#include "WeightPdf.cpp" // C.5.1

    void main
//  =========
    (
    )
    {
      double ** Power;
      InitAntiLog ( );
      InitPower ( Power, One, MaxPower );

      double Weights [MaxCat];
      long   Cases   [MaxCat];

      Cases[0] = 12;  Weights[0] = 1.0;
      Cases[1] =  4;  Weights[1] = 0.3;
      Cases[2] =  0;  Weights[2] = 0.0;

      long NoPts = 2;

      double WgtPdf[MaxProb];
      WeightPdf ( WgtPdf, Power, Cases, Weights, NoPts );

      FreePower ( Power, One );
    }
```

C.5.1 WeightPdf File

```
//                              WeightPdf.cpp
//                              =============
//
```

C.5.1.1 WeightPdf

```
    template <class NUM> void WeightPdf
//  ===================================
    (
      double  Pdf  [MaxProb],
```

```
    NUM   ** Power,
    long     Cases [MaxCat],
    double   Wghts [MaxCat],
    long     NoPts
)
{

    static NUM Wk[2][MaxProb][MaxWgt];
    NUM   Zero = 0.0;

    long In = 0;
    long Rs = 1;

//  Initialise diagonal with first weight of 1.0

    for ( long I=0; I<MaxProb; I++ ) {
      for ( long C=0; C<MaxWgt; C++ ) {
        Wk[In][I][C] = Zero;
      }
      C = (long) ( Wghts[0] * I + 0.5 );
      Wk[In][I][C] = Power[I][Cases[0]];
    }

//  Iterale over rest of the weights

    for ( long N=1; N<=NoPts; N++ ) {
      for ( long I=0; I<MaxProb; I++ ) {
        for ( long C=0; C<MaxWgt; C++ ) {
          Wk[Rs][I][C] = Zero;
        }
      }
      for ( long Dp=0; Dp<MaxProb; Dp++ ) {
        long Dw = (long) ( Wghts[N] * Dp + 0.5 );
        NUM P = Power[Dp][Cases[N]];
        for ( long Org=0; Org<MaxProb-Dp; Org++ ) {
          for ( long Wgt=0; Wgt<MaxWgt-Dw; Wgt++ ) {
            Wk[Rs][Org+Dp][Wgt+Dw] += Wk[In][Org][Wgt] * P;
          }
        }
      }

      long Swap = In;
      In = Rs;
      Rs = Swap;
    }

//  Normalise and gather where probability is one

    NUM Total = Zero;
    for ( long S=0; S<MaxWgt; S++ ) {
      Total += Wk[In][One][S];
    }

    NUM Recip = 1.0 / Total;
    for ( S=0; S<MaxWgt; S++ ) {
      Pdf[S] = Double ( Wk[In][One][S] * Recip );
    }
  }
```

D Collecting Data

A framework is proposed that logically splits a system into small interchangeable components so that each small piece of the new intelligent system can be separately evaluated against the equivalent piece in the old system. As much of the evaluation as possible can then be run as a single blind experiments.

A generalisation of the medical process to be modelled by the intelligent medical system is presented in Figure D.1. A patient is examined by an 'Examining' physician, assisted by appropriate 'Monitor' equipment, and clinical notes ('Present Input') are written. The physician will undoubtedly also make subconscious observations that cannot be recorded ('Subconscious Information'). The same, or a different, 'Cognitising' physician will then form an opinion from the notes. This opinion will be recorded on the notes ('Present Output'). If the physician is the same, he/she will have access to the 'Subconscious Information' from the examination. From the noted opinion, the same, or a different, 'Acting' physician will then take some action, e.g. administer drugs, operate, or discharge the patient. If the 'Acting' physician is the same as the 'Examining' physician, he/she will have access to the 'Subconscious Information'. The action taken will then be recorded in the notes ('Present Actions').

While the above description has been slanted towards a diagnostic and treatment process, it can be applied in general to any 'Expert' process in that the inputs must be gathered, an opinion formed, and then appropriate action taken, all appropriately recorded. It should also be noted that the three phrases are only logically separate, physically the physician may be doing all three in parallel.

Figure D.2 represents substitution of an intelligent medical system for the physician's cognition. Both the monitoring equipment and the physician enter the examination data into an 'Input Module' of the expert system. The 'Expert System' forms an opinion for the physician to act on as before. If the 'Acting' physician is the same as the 'Examining' physician he/she has full access to the input data, and of course any 'Subconscious Information' about the patient. Whether the physician will override the expert system in

cases of differences of opinion is a moot point (section 2.1.3). The action of the physician should then be recorded by an 'Actions Module', in machine readable form, in the expert system database.

Figure D.3 represents the most automated intelligent medical system, where the patient is wired up to a 'Monitor', the 'Intelligent System' forms an opinion, and a 'Robot' acts on it, e.g. infusing drugs.

Note that the expert system has been split into up to five separate components, an 'Input Module', the 'Intelligent System' itself, an 'Output Module', a 'Robot', and an 'Actions Module'. The 'Input', 'Output' and 'Actions' modules should all have an interactive dialogue for inputting information directly, or for entering cases already stored on existing media (e.g. clinical notes on paper), access to the intelligent medical system database for storing them, and a means of displaying/printing the information again. It should be emphasised that the 'Output Module' has the facility for entering the conclusions of cases assessed by human experts, and of printing them out in exactly the same format as cases assessed by the expert system.

D.1 Laboratory Tests

A direct comparison between human experts and the intelligent medical system can therefore be made during laboratory testing by assessing each test case through two different routes. Once each case is entered into the input module, it can either be printed out from the input module, assessed by a human expert, and the conclusions interactively entered into the output module, or the case can be assessed by the intelligent medical system, and written to the output module. Each conclusion can then be printed out in identical format so enabling direct comparison of system and human performance.

It is this interchangeability of components that will allow testing to be more focused on components of the system. The 'Input Module' could be tested by comparing human experts assessing the 'Present Input' and entering their conclusions into the 'Output Module', to human experts assessing printout from the 'Input Module' of previously entered cases, and entering their conclusions into the 'Output Module'. A group of *2E* experts could test

2C cases by the even numbered cases being assessed by the even numbered experts via the Input Module, and the odd cases via the 'Present Input', then reversing the allocation for the odd numbered experts. Another group of experts would then assess the outputs. Significantly greater performance obtained from the 'Present Input' would indicate that not all relevant information is being stored in the 'Input Module'

The output module could be tested by a group of experts being given cases printed out from the 'Input Module' and either presenting their assessment in the 'Present Output' format, or entering, and printing out their assessments from the 'Output Module'. Another group of experts, knowledgeable about the actions normally taken with the 'Present Output', could giving a qualitative comparison. Differences in performance should be investigated to see if they reveal inadequate representation in the 'Output Module'.

Figure D.4 shows how this 'Follow Up' Information can be used to produce a 'Fuzzy Gold Standard' output, by which the system, and human experts can be compared.

A 'Follow up' physician will collect the additional data, hopefully a 'Gold Standard' but possibly a 'Silver Standard' and either record it in machine readable, or non machine readable form. The physician is again likely to pick up information subconsciously. The same, or a different physician will evaluate the case from the initial data (either displayed from the Input module, or on existing media – the two should have identical information), the follow up data, and if he/she did the follow up, from 'Subconscious information'. The 'Evaluating' physician will form a 'Fuzzy Gold Standard' opinion and enter it in the 'Output Module'. A 'Fuzzy Gold Standard' can also be given by a reassessment of the case by the original 'Examining medic' once the follow up information is available. The original 'Examining', 'Follow up', and 'Evaluating' physician can all be the same person, as long as access to 'Subconscious Information' is logged on the database.

D.2 Field Trial

Once laboratory testing has been completed and the system is seen as safe by the 'experts', a Field trial can be conducted.

If an expert system is to deliver expert performance, the significance of the 'Subconscious Information' probably stored in the head of the 'Examining' physician needs to be measured. This can be done by one physician entering the data into the input module in the 'live' situation, and then entering the assessment of that data into the 'Output Module', compared to another assessing the same case just from the output of the 'Input Module' as entered by the first physician.

It is probably not relevant to measure the impact of 'Subconscious Information' on the 'Acting Medic', as the situations where the 'Acting Medic' is the same as the 'Examining Medic' are not likely to change with the introduction of an expert system.

D.3 Direct Action

If the intelligent system is going to take direct action e.g. infuse drugs, the proposed architecture can be used to test this part.

The system runs from 'Monitor' to 'Output Module', but uses the 'Input Module' to give a full display to the 'Acting' physician of the 'Monitor' output. The 'Acting' physician thus acts as the 'Examining' and 'Cognitising' physician as well. The 'Acting' physician reads the output from the 'Output Module' and acts upon it only when he/she considers it safe to do so. The virtual actions of the 'Robot' and the actual actions of the 'Acting' physician are both recorded by the 'Action Module'.

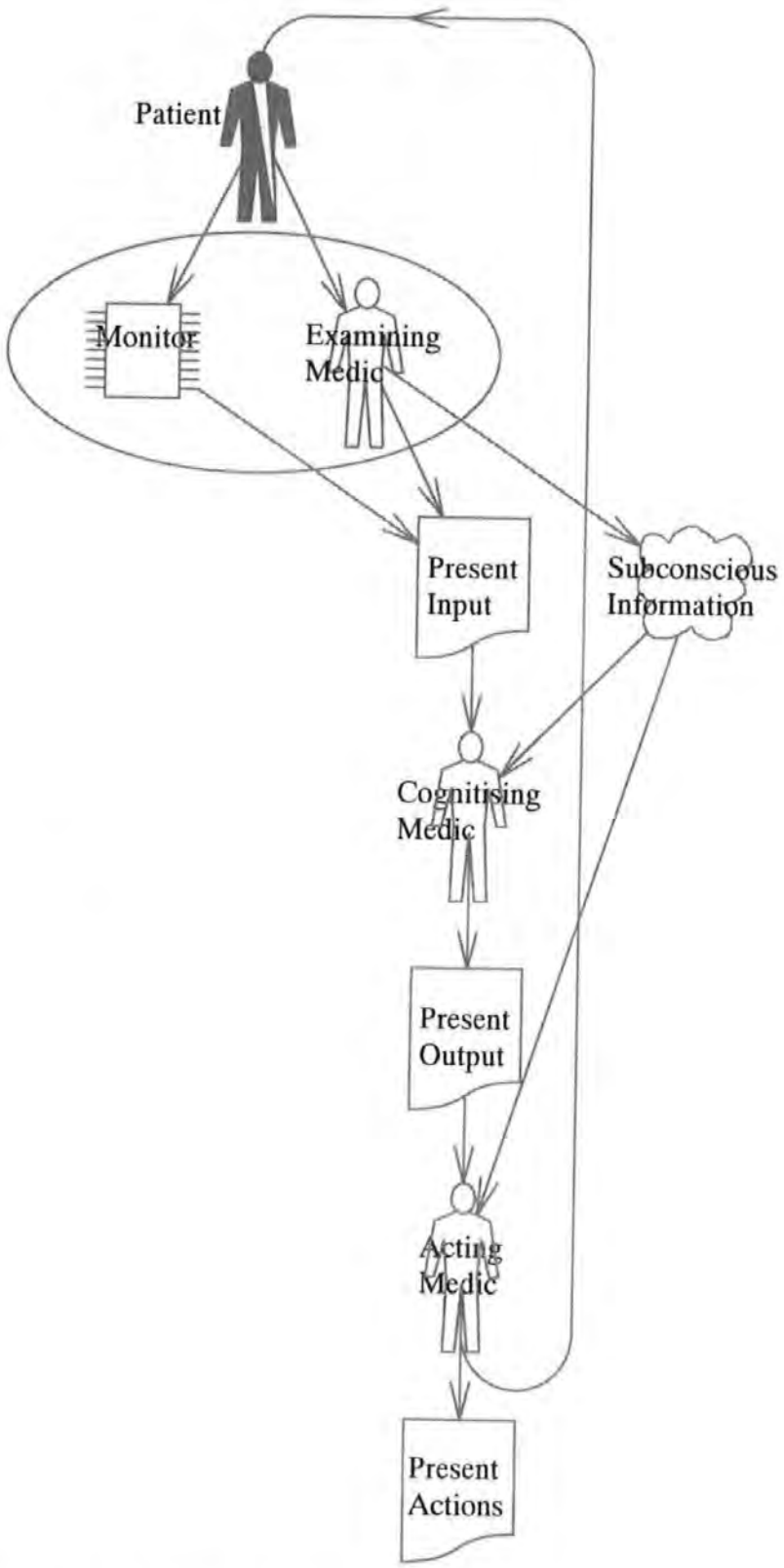


Figure D.1 Flow of information through current medical system

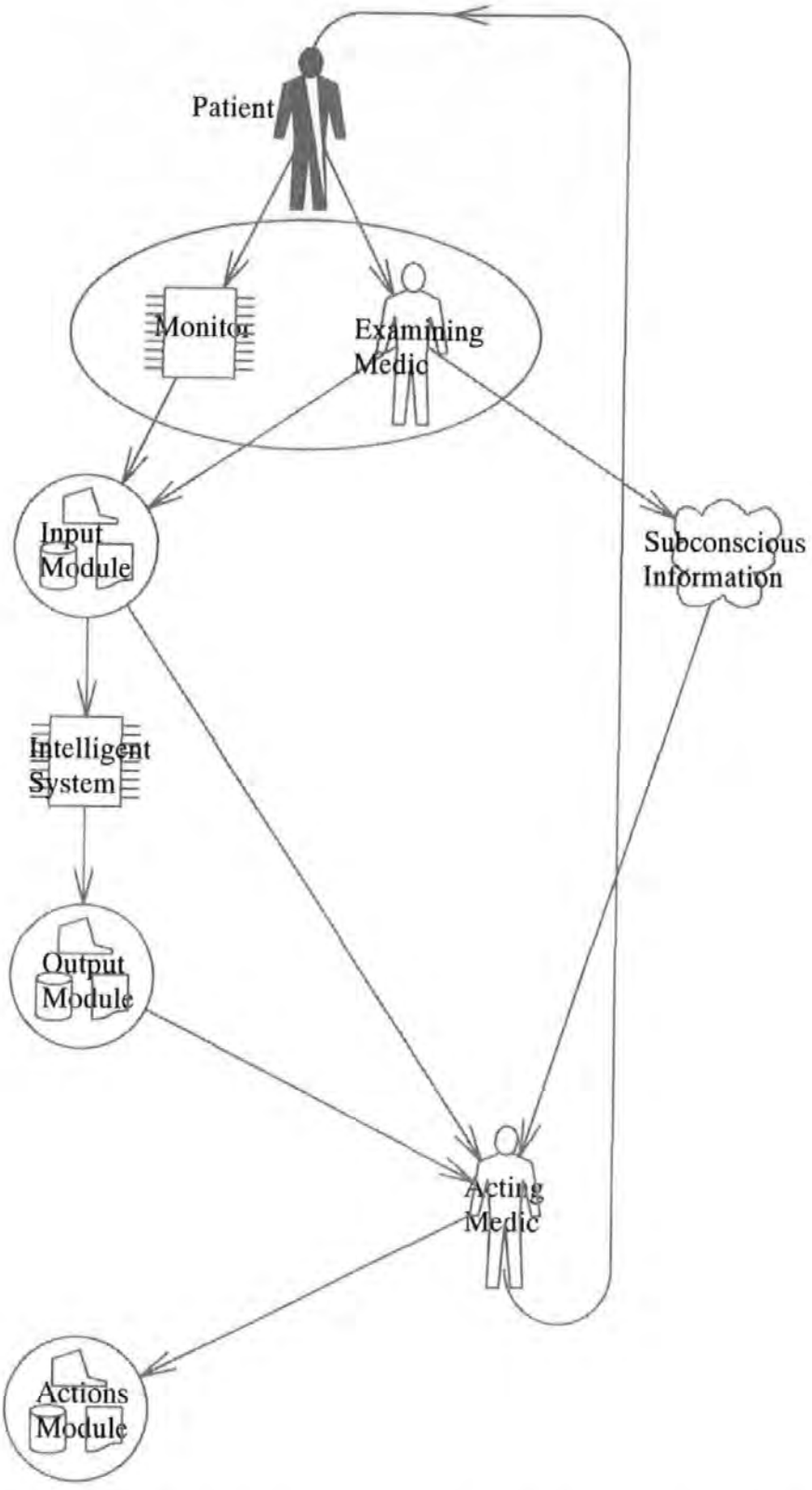


Figure D.2 Flow of information through an intelligent medical system

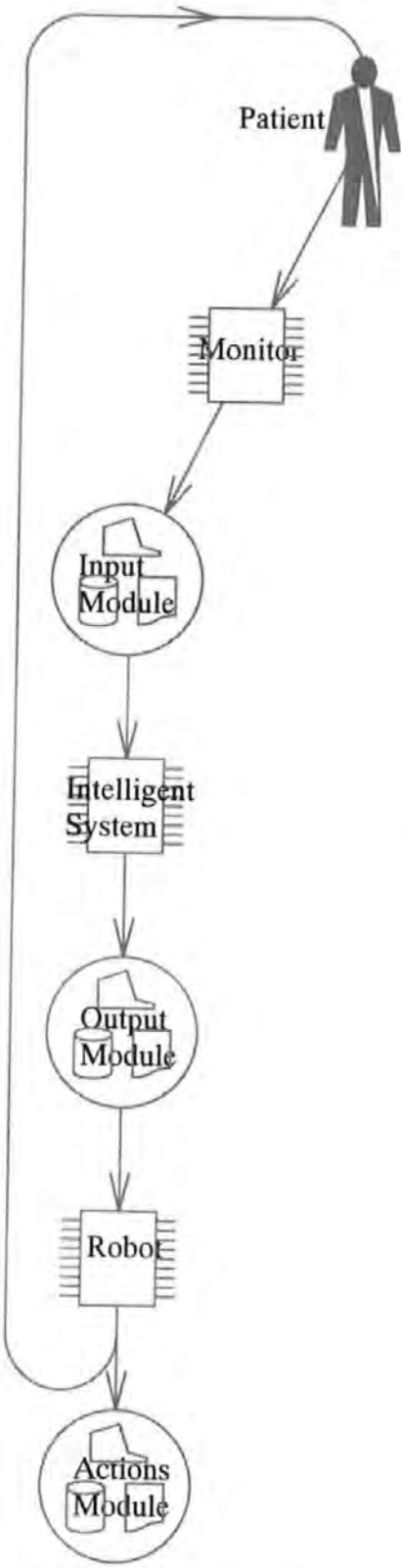


Figure D.3 Flow of information through automated intelligent system

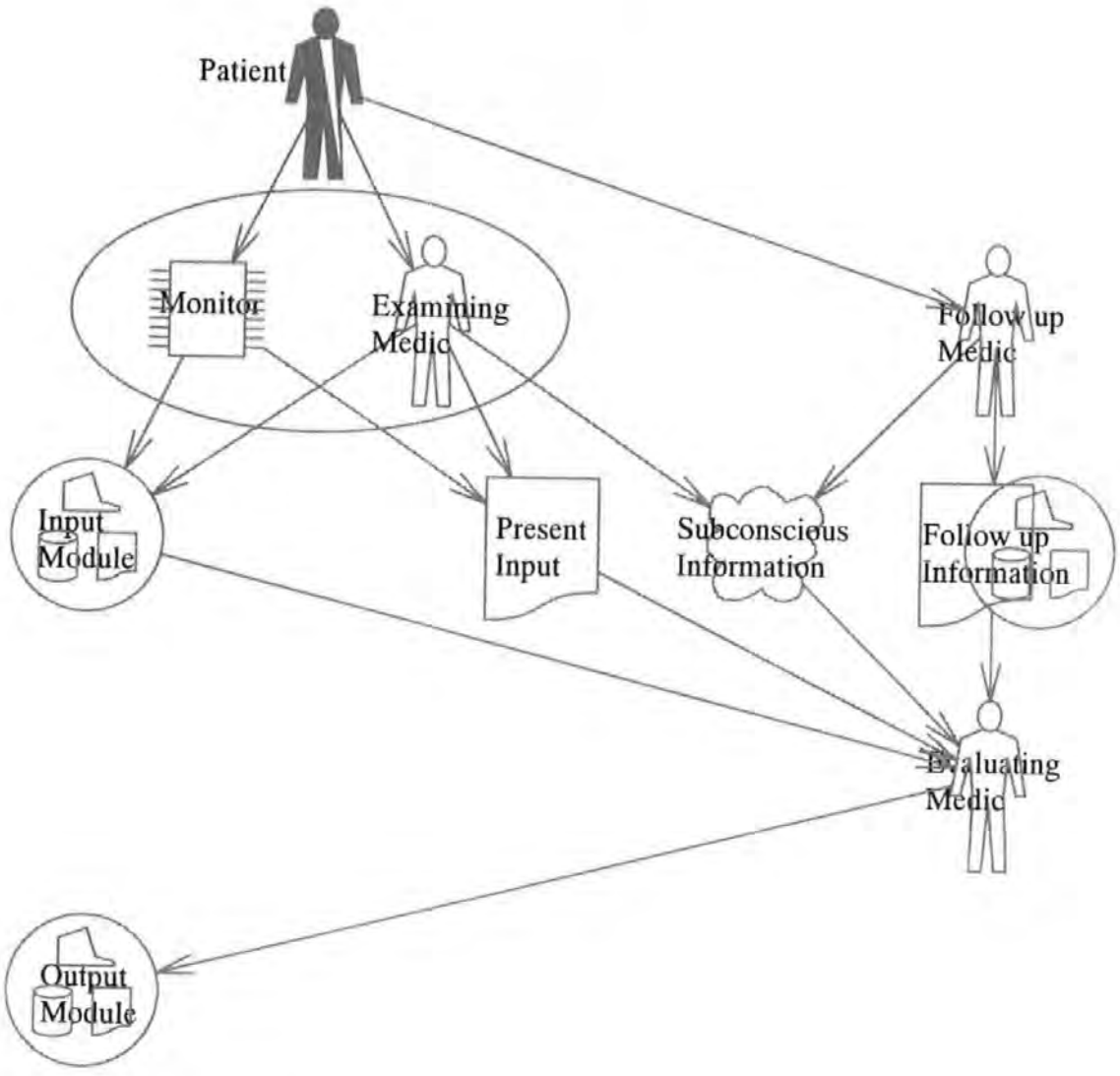


Figure D.4 Flow of evaluation information through system

E Published Paper

Receiver Operating Characteristic Analysis for Intelligent Medical Systems—A New Approach for Finding Confidence Intervals

Julian B. Tilbury, Peter W. J. Van Eetvelt, Jonathan M. Garibaldi, John S. H. Curnow, and Emmanuel C. Ifeachor*

Abstract—**Intelligent systems are increasingly being deployed in medicine and healthcare, but there is a need for a robust and objective methodology for evaluating such systems. Potentially, receiver operating characteristic (ROC) analysis could form a basis for the objective evaluation of intelligent medical systems. However, it has several weaknesses when applied to the types of data used to evaluate intelligent medical systems. First, small data sets are often used, which are unsatisfactory with existing methods. Second, many existing ROC methods use parametric assumptions which may not always be valid for the test cases selected. Third, system evaluations are often more concerned with particular, clinically meaningful, points on the curve, rather than on global indexes such as the more commonly used area under the curve.**

A novel, robust and accurate method is proposed, derived from first principles, which calculates the probability density function (pdf) for each point on a ROC curve for any given sample size. Confidence intervals are produced as contours on the pdf. The theoretical work has been validated by Monte Carlo simulations. It has also been applied to two real-world examples of ROC analysis, taken from the literature (classification of mammograms and differential diagnosis of pancreatic diseases), to investigate the confidence surfaces produced for real cases, and to illustrate how analysis of system performance can be enhanced. We illustrate the impact of sample size on system performance from analysis of ROC pdf's and 95% confidence boundaries. This work establishes an important new method for generating pdf's, and provides an accurate and robust method of producing confidence intervals for ROC curves for the small sample sizes typical of intelligent medical systems. It is conjectured that, potentially, the method could be extended to determine risks associated with the deployment of intelligent medical systems in clinical practice.

Index Terms—**Confidence interval, intelligent medical system, performance evaluation, probability density function, receiver operating characteristic (ROC).**

Manuscript received July 22, 1999; revised March 2, 2000. The work of J. Tilbury was supported by the EPSRC under a research studentship. *Asterisk indicates corresponding author.*

J. B. Tilbury, P. W. J. Van Eetvelt, and J. M. Garibaldi are with the SMART Medical Systems Research Group, School of Electronic Communication and Electrical Engineering, University of Plymouth, Drake Circus, Plymouth PL1 8AA, U.K.

J. S. H. Curnow is with the Department of Medical Physics, Derriford Hospital, Derriford, Plymouth PL8 6DH, U.K., and the SMART Medical Systems Research Group, School of Electronic Communication and Electrical Engineering, University of Plymouth, Drake Circus, Plymouth PL1 8AA, U.K.

*E. C. Ifeachor is with the SMART Medical Systems Research Group, School of Electronic Communication and Electrical Engineering, University of Plymouth, Drake Circus, Plymouth PL1 8AA, U.K. (e-mail: e.ifeachor@plymouth.co.uk).

Publisher Item Identifier S 0018-9294(00)05136-3.

I. Introduction

INTELLIGENT systems are increasingly being deployed in medicine and healthcare, to practically aid the busy clinician and to improve the quality of patient care [1]–[7]. The need for an objective methodology for evaluating such systems is widely recognized [2], [4], [8]–[12]. In medicine and healthcare, where safety is critical, this is important if techniques such as medical expert systems and neural systems are to be widely accepted in clinical practice.

The work described here, arose from a critical investigation into the potential role of receiver operating characteristic (ROC) analysis as a basis for objective evaluation of intelligent medical systems. The work forms part of an initiative to develop a theoretical framework for an objective methodology in evaluating intelligent systems in this safety critical area.

ROC analysis is now common in medicine and healthcare [4], [13], [14], particularly in radiology [8], [10], [15], where it is used to quantify the accuracy of diagnostic tests [10], [16], [17]. The performance of an "expert," human or machine, can be represented objectively by ROC curves [10], [18]. Such curves show, for example, the tradeoff between a diagnostic test correctly identifying diseased patients as diseased, rather than healthy, versus correctly identifying healthy patients as healthy, rather than diseased. Many intelligent medical systems carry out this type of task, whether actually classified as "diagnostic" [19], or not, e.g., "prognostic" [20].

To serve as a basis for objective evaluation of intelligent medical systems, ROC analysis will need to be extended to address a number of limitations. In practice, ROC analysis can be either parametric or nonparametric. In parametric analysis, the underlying population distributions of the diseased and healthy patients are often assumed to be normal. However, other types of distributions, such as gamma and negative exponential, are sometimes used. In contrast, nonparametric analysis does not make any assumptions about the form of the underlying population distributions. When an intelligent medical system is tested against human experts, a small number of cases are picked, often by an independent expert. If the underlying population distributions of diseased and healthy patients are normal, a random sample of patients from the population would preserve these distributions. However, if the cases are picked by an independent expert, particularly one who is deliberately biased in favor of picking difficult cases, the distribution may not be preserved. Thus, nonparametric methods may be more appropriate for intelligent medical system testing.

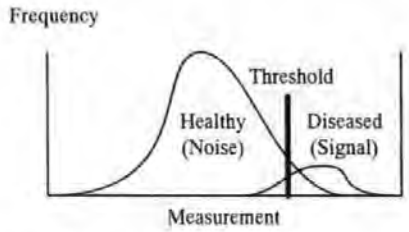


Fig. 1. The underlying model for ROC curves.

ROC curves are a complex representation of performance, and for convenience, the area under the ROC curve (AUC) is used as a single index of accuracy. Intuitively, the area under the ROC curve gives the probability of correctly identifying a healthy patient from a pair where one is known to be diseased and the other one is known to be healthy [21]. However, an intelligent medical system is more likely to be required to make a decision about the disease status of a single patient of unknown health, which makes the AUC of limited practical value in the clinical situation.

As human experts are available for a limited time, the number of cases used for evaluation is often small. Existing methods of ROC analysis are often unsatisfactory with small numbers of cases, especially if the AUC is high. Under these circumstances their confidence intervals can be erroneous, or the algorithms can even fail to produce any result. Obuchowski and Lieber [22] compared 11 methods of parametric and nonparametric analysis and were unable to find a single best alternative for constructing the confidence interval, when the sample size was small.

The aim of the work reported here was to find a nonparametric method of ROC analysis that would be robust and accurate over any sample size, particularly small samples, and could be applied to particular points on the ROC curve of clinical interest. To achieve this, the underlying probability theory was re-examined, and a novel method of producing a probability density function (pdf) over the whole ROC graph, for each point on the curve, was derived. The theoretical work has been validated by Monte Carlo simulations, and it has also been applied to two real-world examples taken from the literature. While many Monte Carlo simulations assume a fixed population ROC curve and examine the distribution of samples generated randomly from the fixed population, our simulation generated random samples from random population ROC curves.

II. ROC Curves

Taking the situation where there are two types of events, signal (diseased), and noise (healthy), it is hoped to distinguish between them by measuring a characteristic property of these events, on an ordinal, interval or ratio scale. Fig. 1 gives a hypothetical example of the relative frequency with which two types of events give different values of the measured property. To distinguish between the types of event, a threshold is chosen so that events with a measurement lower than the threshold are labeled as noise, and events with a measurement greater than the threshold are labeled as signal. Since the two distributions overlap, no threshold value will completely separate them. Table I shows the 2×2 contingency table of the actual type of an event, against its test classification according to the threshold.

TABLE I
SINGLE THRESHOLD CONTINGENCY

		Standard: Signal	Standard: Noise
Test	Signal	True Positive b_0	False Positive a_0
	Noise	False Negative b_1	True Negative a_1

TABLE II
THREE THRESHOLD CONTINGENCY, WHERE ϕ IS THE MEASUREMENT

		Standard: Signal	Standard: Noise
Test	$\phi \geq$ Threshold 0	b_0	a_0
	Threshold 0 $> \phi \geq$ Threshold 1	b_1	a_1
	Threshold 1 $> \phi \geq$ Threshold 2	b_2	a_2
	Threshold 2 $> \phi$	b_3	a_3

This table assumes there is a standard by which the actual type of the event is known.

The test can then be characterized by two ratios:

$$\text{Hit Rate} = \frac{\text{True positive}}{\text{True positive} + \text{False negative}} = \frac{b_0}{b_0 + b_1}$$

$$\text{False Alarm Rate} = \frac{\text{False positive}}{\text{False positive} + \text{True negative}} = \frac{a_0}{a_0 + a_1}$$

If multiple thresholds are used, for example, to categorise events into "definitely signal," "possibly signal," "possibly noise" and "definitely noise," the contingency table can be expanded to a $2\times n + 1$ table, where n is the number of thresholds; for example, to a 2×4 table, as shown in Table II.

From the table, n pairs of Hit Rate and False Alarm Rate can be calculated. For example, Table II gives the following three pairs:

$$\text{Hit Rate}_0 = \frac{b_0}{b_0 + b_1 + b_2 + b_3}$$

$$\text{False Alarm Rate}_0 = \frac{a_0}{a_0 + a_1 + a_2 + a_3},$$

$$\text{Hit Rate}_1 = \frac{b_0 + b_1}{b_0 + b_1 + b_2 + b_3}$$

$$\text{False Alarm Rate}_1 = \frac{a_0 + a_1}{a_0 + a_1 + a_2 + a_3},$$

$$\text{Hit Rate}_2 = \frac{b_0 + b_1 + b_2}{b_0 + b_1 + b_2 + b_3}$$

$$\text{False Alarm Rate}_2 = \frac{a_0 + a_1 + a_2}{a_0 + a_1 + a_2 + a_3}.$$

With a sufficiently large number of thresholds changing in small discrete steps, a plot of Hit Rate (along the y axis) against

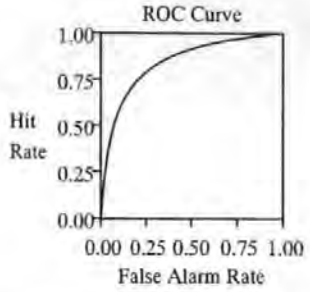


Fig. 2. Example ROC curve.

False Alarm Rate (along the x axis) for each threshold gives a ROC curve. A typically shaped curve for a multi threshold plot is given in Fig. 2.

The curve thus shows the tradeoff between correctly detecting a signal, and mistaking noise for a signal. If the two underlying population distributions are well separated, the curve will immediately rise to the top left corner (0.0, 1.0), and then proceed horizontally. If the distributions tend to overlap, so that noise and signal cannot be distinguished by the measurement, the curve will approach the diagonal (0.0, 0.0 to 1.0, 1.0).

If a ROC curve is plotted for a sample of cases, the curve will only be an estimate of the actual ROC curve of the population. Confidence intervals, therefore, need to be given. Many methods of producing confidence intervals for the AUC, both parametrically [23] and nonparametrically [17], [18], [21], [22], [24], [25], and for each individual point on a nonparametric curve [26]–[28] have been given. However, for small samples sizes, typical in intelligent medical system testing, none of these methods is ideal [22]. In the case of binormal parametric models, the methods can fail to produce any results at all when the signal (diseased) and noise (healthy) samples do not overlap. This can happen particularly with small samples, when the population AUC approaches 1.0 [29].

III. A New Approach to ROC Analysis

The method proposed here, assumes a nonparametric model and is robust and accurate over all data sets. It returns to the underlying probability theory, to construct a pdf over the entire ROC graph for each point of the curve. The method is based upon asking the following question for every possible point on the surface of the ROC graph:

"If this point represents the true Hit Rate and False Alarm Rate of the population, what would be the probability of getting the sample actually obtained?"

If that question can be answered for every point on the graph, and normalized so that the total probability of every point on the surface sums to one, a pdf for the true Hit Rate and False Alarm Rate can be generated. By dividing the surface into a fine grid, and integrating the expression for the probability of every point over each square of the grid, the surface can then be presented as a three-dimensional mesh, or contour lines can be drawn to enclose an arbitrary percentage of the probability, e.g., 95% of the probability, which gives the 95% confidence interval for the location of the true Hit Rate and False Alarm Rate.

Consider the full situation where:

- y is the Hit Rate of the population, given as a probability;
- x is the False Alarm Rate of the population, given as a probability;
- f is the frequency of disease events in the population, given as a probability;
- b_0 is the number of true positives in the sample;
- a_0 is the number of false positives in the sample;
- b_1 is the number of false negatives in the sample;
- a_1 is the number of true negatives in the sample.

Then, P, the probability of a ROC point being at the location (x, y), is given by the product of three terms. The first term, is the probability of obtaining $b_0 + b_1$ diseased cases in $a_0 + a_1 + b_0 + b_1$ cases when the probability of disease is f. The second term, is the probability of obtaining a_0 False Alarms in $a_0 + a_1$ healthy cases when the probability of a False Alarm is x; and the third term, is the probability of obtaining b_0 Hits in $b_0 + b_1$ diseased cases when the probability of a Hit is y

$$P_{xyfb_0a_0b_1a_1} = \binom{a_0 + a_1 + b_0 + b_1}{b_0 + b_1} f^{b_0+b_1}(1-f)^{a_0+a_1} \cdot \binom{a_0 + a_1}{a_0} x^{a_0}(1-x)^{a_1} \cdot \binom{b_0 + b_1}{b_0} y^{b_0}(1-y)^{b_1}.$$

In order to normalize the probability at each point (x, y), to sum to 1.0 when integrated across the whole surface, the probability is divided by the integral over the surface as shown in (1) at the bottom of the page.

$$\begin{aligned}\text{PointProbability}_{xy} &= \frac{\binom{a_0 + a_1 + b_0 + b_1}{b_0 + b_1} f^{b_0+b_1}(1-f)^{a_0+a_1} \binom{a_0 + a_1}{a_0} x^{a_0}(1-x)^{a_1} \binom{b_0 + b_1}{b_0} y^{b_0}(1-y)^{b_1}}{\binom{a_0 + a_1 + b_0 + b_1}{b_0 + b_1} f^{b_0+b_1}(1-f)^{a_0+a_1} \binom{a_0 + a_1}{a_0} \int_0^1 x^{a_0}(1-x)^{a_1}\,dx \binom{b_0 + b_1}{b_0} \int_0^1 y^{b_0}(1-y)^{b_1}\,dy} \\ &= \frac{x^{a_0}(1-x)^{a_1}}{\int_0^1 x^{a_0}(1-x)^{a_1}\,dx} \frac{y^{b_0}(1-y)^{b_1}}{\int_0^1 y^{b_0}(1-y)^{b_1}\,dy} \end{aligned} \quad (1)$$

Using the *Beta* function to substitute for the integrals (see Appendix for details)

$$\text{PointProbability}_{xy} = \frac{x^{a_0}(1-x)^{a_1}}{\frac{a_0!a_1!}{(a_0+a_1+1)!}} \frac{y^{b_0}(1-y)^{b_1}}{\frac{b_0!b_1!}{(b_0+b_1+1)!}}. \quad (2)$$

To represent the surface, it is divided into a fine grid and the probability of each quantized grid square is calculated by integrating the probability at a point, over the area of each grid square. The integral over the area is equal to the product of two one-dimensional (1-D) integrals along the Hit Rate and False Alarm Rate axes. Therefore, two vectors, $\boldsymbol{X}$ and $\boldsymbol{Y}$, each with i elements, are defined to hold the 1-D integrals

$$\boldsymbol{X}_i = \frac{\int_{(i-1)/n}^{i/n} x^{a_0}(1-x)^{a_1}\,dx}{\frac{a_0!a_1!}{(a_0+a_1+1)}} \quad (3)$$

and

$$\boldsymbol{Y}_i = \frac{\int_{(i-1)/n}^{i/n} y^{b_0}(1-y)^{b_1}\,dy}{\frac{b_0!b_1!}{(b_0+b_1+1)}}$$

$$\text{For all } i \text{ from } i = 1 \text{ to } i = n. \quad (4)$$

The pdf, quantized as a fine grid, is therefore the product of the two vectors

$$\text{Surface} = \boldsymbol{X} \cdot \boldsymbol{Y}^T. \quad (5)$$

Integrating (3) and (4) (see Appendix for details) gives (6) and (7), shown at the bottom of the page.

IV. Multiple Points

The analysis can now be expanded to the general case of multiple ROC points. Since it has been shown above, that the surface can be treated as a product of two probability density vectors, one for Hit Rate and the other for False Alarm Rate, this discussion will examine only one vector, the $\boldsymbol{X}'$, or False Alarm Rate vector, the identical method being applicable to the $\boldsymbol{Y}'$, or Hit Rate Vector.

For a ROC curve of n points, there are $n+1$ classifications of events (threshold ranges). Let there be a_i occurrences of event e_i, where $i = 0, \ldots, n$ (see Table II). Let the true probability of event e_i be x_i. Now, an extension of the hypothesis stated above can be applied, by asking the following question, for every point:

> *"If this point, $x_0, x_1, \ldots, x_n$, represents the true probability of events, $e_0, e_1, \ldots, e_n$, in the population, what would be the probability of getting the actual results $a_0, a_1, \ldots, a_n$ obtained?"*

If that question can be answered for every point, and normalized such that the total probability of every point in the hyper-volume sums to 1, the pdf in $n+1$-dimensional ($n+1$-D) space could be calculated.

By multinomial from the numerator of (1) for the single ROC point case, the probability P', of a point lying on a hyperplane in $n+1$-D space can be written as

$$P' \propto \prod_{i=0}^{n-1} x_i^{a_i} \left(1 - \sum_{i=0}^{n-1} x_i\right)^{a_n}$$

where

$$\sum_{i=0}^{n} x_i = 1$$

To represent the pdf on a two-dimensional (2-D) ROC graph, the $n+1$-D pdf must be mapped into one dimension, i.e., to a $\boldsymbol{X}'$ vector for the False Alarm Rate, or a $\boldsymbol{Y}'$ vector for the Hit Rate, and the 2-D ROC surface formed as the product of the $\boldsymbol{X}'$ and $\boldsymbol{Y}'$ vectors.

Note that each point on the ROC curve represents a different combination of events. The first point represents e_0 events only, but the second point represents the e_0 plus the e_1 events, the third point e_0, e_1 plus e_2 events, and so on. This adds a subtlety to the way the hyperplane is mapped to the linear pdf for each point, which is shown for a four-point ROC curve below.

$$\boldsymbol{X}_i = (a_0+a_1+1)! \sum_{k=0}^{a_1} \frac{\left(\frac{i}{n}\right)^{a_0+a_1+1-k}\left(1-\frac{i}{n}\right)^k - \left(\frac{i-1}{n}\right)^{a_0+a_1+1-k}\left(1-\frac{i-1}{n}\right)^k}{k!(a_0+a_1+1-k)!} \quad (6)$$

$$\boldsymbol{Y}_i = (b_0+b_1+1)! \sum_{k=0}^{b_1} \frac{\left(\frac{i}{n}\right)^{b_0+b_1+1-k}\left(1-\frac{i}{n}\right)^k - \left(\frac{i-1}{n}\right)^{b_0+b_1+1-k}\left(1-\frac{i-1}{n}\right)^k}{k!(b_0+b_1+1-k)!}$$

$$\text{For all } i \text{ from } i = 1 \text{ to } i = n \quad (7)$$

Let s_0 be the actual probability of the first ROC point. The first point represents only the true probability of the c_0 events. As s_0 varies from zero to one, it is directly related to x_0 by the relationship $s_0 = x_0$. The hyperplane is mapped into the line by integrating across slices at right angles to the x_0 axis.

Let s_1 be the actual probability of the second ROC point. The second point is the combined probability of the c_0 events, plus c_1 events, in the population. If does not matter what the individual probability of c_0 events is, or what the individual probability of c_1 events is, only the combined probability matters. As s_1 varies from zero to one, x_0 and x_1 are constrained by the relation $s_1 = x_0 + x_1$. The hyperplane is mapped into the line by integrating across 2-D diagonal slices at right angles to the line $x_0 + x_1 = 1$.

Similarly, let s_2 be the actual probability of the third ROC point. The third point is the combined probabilities of c_0, c_1 and c_2 events. As s_2 varies from zero to one, x_0, x_1 and x_2 are constrained by the relationship $s_2 = x_0 + x_1 + x_2$. The hyperplane is mapped into the line by integrating across the 3-D diagonal slices at right angles to the line $x_0 + x_1 + x_2 = 1$.

By way of example, the integrals for a four point ROC curve are as follows.

A. Four ROC Points, First Point

$$f(s_0) = \int_0^{1-s_0} \int_0^{1-s_0-x_1} \int_0^{1-s_0-x_1-x_2} s_0^{a_0} x_1^{a_1} x_2^{a_2} x_3^{a_4} \cdot (1 - s_0 - x_1 - x_2 - x_3)^{a_4}\, dx_3\, dx_2\, dx_1. \quad (8)$$

The variable s_0 ranges from 0–1 across the pdf. In this case s_0 is equivalent to x_0. Since the function is constrained to the hyperplane $x_0+x_1+x_2+x_3+x_4 = 1$, x_1 is thus confined to the range $1 - s_0$, which are therefore the limits of the outer integral. Similarly, x_2 is then confined to the range 0 to $1 - s_0 - x_1$, the limits of the middle integral, and x_3 is confined to the range 0 to $1 - s_0 - x_1 - x_2$, the limits of the inner integral. The expression is kept on the hyperplane by substituting $x_4 = 1 - s_0 - x_1 - x_2 - x_3$, in the last term.

Integrating (8) (see Appendix for details) gives:

$$f(s_0) = s_0^{a_0}(1 - s_0)^{a_1+a_2+a_3+a_4+3} \cdot \frac{a_1!a_2!a_3!a_4!}{(a_1 + a_2 + a_3 + a_4 + 3)!}. \quad (9)$$

B. Four ROC Points, Second Point

$$f(s_1) = \int_0^{s_1} \int_0^{1-s_1} \int_0^{1-s_1-x_2} x_0^{a_0}(s_1 - x_0)^{a_1} x_2^{a_2} x_3^{a_7} \cdot (1 - s_1 - x_2 - x_3)^{a_4}\, dx_3\, dx_2\, dx_0$$

Again, the variable s_1 ranges from zero to one across the pdf. In this case, $s_1 = x_0 + x_1$, and therefore x_0 is confined to the range zero to s_1, which are therefore the limits of the outer integral. In the second term $s_1 - x_0$ is substituted for x_1, and, in the last term, s_1 is substituted for $-(x_0 + x_1)$. The range of x_2 is then confined to the range zero to $1 - s_1$, the limits of the middle integral, and x_3 is confined to the range zero to $1 - s_1 - x_1$, the limits of the inner integral.

$$f(s_1) = s_1^{a_0+a_1+1} \frac{a_0!a_1!}{(a_0 + a_1 + 1)!}(1 - s_1)^{a_2+a_3+a_4+2} \cdot \frac{a_2!a_3!a_4!}{(a_2 + a_3 + a_4 + 2)!}.$$

C. Four ROC Points, Third Point

$$f(s_2) = \int_0^{s_2} \int_0^{s_2-x_0} \int_0^{1-s_2} x_0^{a_0} x_1^{a_1} (s_2 - x_0 - x_1)^{a_2} x_3^{a_3} \cdot (1 - s_2 - x_3)^{a_4}\, dx_3\, dx_1\, dx_0.$$

Here, $s_2 = x_0 + x_1 + x_2$. This is used in the third and fifth term. The outer and middle integrals are limited by this expression. The inner integral is constrained by the hyperplane

$$f(s_2) = s_2^{a_0+a_1+a_2+2} \frac{a_0!a_1!a_2!}{(a_0 + a_1 + a_2 + 2)!}(1 - s_2)^{a_3+a_4+1} \cdot \frac{a_3!a_4!}{(a_3 + a_4 + 1)!}.$$

D. Four ROC Points, Fourth Point

$$f(s_3) = \int_0^{s_3} \int_0^{s_3-x_0} \int_0^{s_3-x_0-x_1} x_0^{a_0} x_1^{a_1} x_2^{a_2} \cdot (s_3 - x_0 - x_1 - x_2)^{a_3}(1 - s_3)^{a_4}\, dx_2\, dx_1\, dx_0.$$

Here, $s_3 = x_0 + x_1 + x_2 + x_3$. This is used in the fourth and fifth term. All the integrals are limited by this expression. The hyperplane constraint is only evident in the last term

$$f(s_3) = s_3^{a_0+a_1+a_2+a_3+3} \frac{a_0!a_1!a_2!a_3!}{(a_0 + a_1 + a_2 + a_3 + 3)!}(1-s_3)^{a_4}.$$

E. The General Result for Multiple ROC Points

When any $f(s_n)$ is normalized by dividing it by its integral, as shown for the one ROC point example in (1), the factorial terms cancel out. Integration of the expressions for each point of one, two, three, and four point ROC curves reveals a pattern, which by induction generalizes to

$$f(s) = s^{\sum_{i=0}^{n-1} a_i + n - 1}(1 - s)^{\sum_{j=n}^{m} a_j + m - n}$$

where n is the number of the point, and m the total number of points in the curve.

If

$$a_0' = \sum_{i=0}^{n-1} a_i + n - 1$$

and

$$a_1' = \sum_{j=n}^{m} a_j + m - n.$$

The multipoint ROC equations are in exactly the same form as the numerator of the single-point ROC equation (1) and since the denominator is the intergal over the whole hyper-volume used

to normalize the distribution to sum to 1.0, the same method can be applied to calculate the pdf.

It should be noted that the pdf of n ROC points actually exists in $2n$-dimensional ($2n$-D) space. The mapping to n 2-D probability surfaces, overlaid on one ROC curve, is merely a convenient representation of this single multidimensional pdf.

V. Plotting the Surface

A computer program was written in C++, to perform the calculation, and plot the 95% confidence interval contour. The surface was quantized to 256×256 elements for the graphical presentation. The actual code optimized the mathematical expression (6) [and (7)] by only calculating, and storing in a vector, the boundary values

$$\text{Boundary Value} = (a_0 + a_1 + 1)! \sum_{k=0}^{a_1} \frac{(\frac{i}{n})^{a_0+a_1+1-k}(1-\frac{i}{n})^k}{k!(a_0+a_1+1-k)!}$$

Each element was then calculated as the difference between two boundary values. Because of the range of the exponent required in the calculation, the excess exponent was held in a long integer as each value was calculated. Many terms in the expression were precalculated and accessed from look-up tables.

The surface was then calculated by a product of the two vectors. The "tiles" of the surface were then sorted by probability, largest first, and marked in order, from the largest, as being inside the confidence boundary until the sum of the marked "tiles" equaled $z\%$ of the sum of all the tiles. A boundary drawing algorithm was then applied, to draw around the marked area to give the $z\%$ confidence boundary.

VI. Simulation Study

To validate the method and the algorithm, a Monte Carlo simulation was performed. The L'Ecuyer [30] pseudorandom number generator, with a period in excess of 2×10^{18}, incorporating a Bays–Durham shuffle with added safeguards was used. Samples with 2^n, $n = 0 \cdots 10$ cases were simulated. Each sample size was simulated with a different set of frequencies of disease in the population. Samples with one case, were simulated with a frequency of disease of 1/2, samples with two cases with frequencies of disease 1/2 and 1/4, through to samples with 1024 cases being simulated with frequencies of $1/2^n$, $n = 1 \cdots 11$. It was not considered worthwhile simulating situations where the frequency of disease, in relation to the number of cases, would often result in no diseased cases at all. For each sample size, at each frequency of disease, ROC curves with 1, 2, 4, 8, and 16 points were simulated. Each point of the multi point ROC curves were simulated independently of the other points. This was to avoid correlation effects, as each 2-D pdf is a different view of the same multidimensional pdf of all the ROC points combined. For each test, a population Hit Rate and False Alarm Rate were generated for all points, whatever the actual point under test. Then each event within the sample, was randomly assigned to the diseased or healthy groups, according to the frequency of disease in the population, and categorized according to the previously generated population Hit Rates and False Alarm Rates. For instance, a four-point ROC curve has five categories. This synthesized data was used to generate the pdf of each sample, for the chosen point. The position of the actual population ROC point, within the pdf was then recorded, and used to produce a histogram of 20 bins, giving the number of times the points fell in the 5%, 10%, ..., 95%, 100% confidence interval. Each test was run 2000 times, with the expectation that about 100 points would be found in each of the 20 confidence intervals. A chi-squared (χ^2) measure was taken of the 2000 tests, and the experiment repeated 200 times to obtain a histogram of the chi-squared values. The total simulation thus generated $(1+2+\cdots+11)\times(1+2+4\cdots+16)$ chi-squared histograms of 200×2000 simulated ROC curves, and ran for over a month on a powerful Unix workstation. If the experiment was working, each histogram of chi-squares would approximate the chi-squared distribution for 19 degrees of freedom. The results of the Monte Carlo simulation are discussed in Section VII.

The population Hit Rate and False Alarm Rate for multi point ROC curves were produced by generating a uniformly distributed random number between zero and one for the population Hit Rate of each point and sorting them into ascending order. The same was done for the False Alarm Rate. The Hit Rates and False Alarm Rates were then paired together in the sorted order. This produced data compatible with the ROC curve format, but without parametric assumptions.

It should be noted that the simulation study is unusual in the method of picking the population ROC curves. Many studies [17], [23], [31] use the following procedure:

- fix the parameters of the curve, e.g., to a binormal curve with an AUC of 0.8;
- generate random data samples from that population curve;
- generate sample ROC curves from the data;
- verify that the confidence limits (e.g., 95%) of the sample curves, contains the population curve, the correct percentage of the time.

The current study used the following procedure:

- simulate population ROC points occurring anywhere on the surface;
- generate data samples from the point;
- plot the pdf from the data sample;
- verify that the point was within given percentiles of the pdf the correct percentage of the time.

The method used for this study will not work when the parameters of the curve are fixed. This can be explained by considering the following Gedankenexperiment. Fig. 3 shows the pdf of a ROC point as a contour map, with the 33%, 66%, and 100% contour marked. The 100% contour covers the whole graph. The interpretation of the contours, is that 33% of the population points that might have produced this sample, are inside the 33% contour, 33% of the points are between the 33% and the 66% contour, and 34% are between the 66% and 100% contour. For the sake of the Gedankenexperiment, the contours should be regarded as steps with uniform density within each contour. Now consider running a Monte Carlo experiment, where the sample that produced this pdf happens

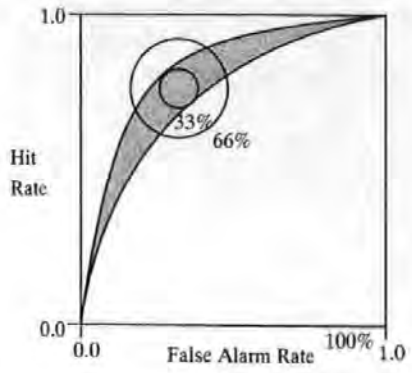


Fig. 3. Gedankenexperiment on the distribution of the ROC curves tested.

to be generated 100 times. If the contours are correct, about 33, 33, and 34 population points will land within each contour respectively. Now consider only drawing the test population ROC points from the grey area which shows a region of hypothetical ROC curves. The easiest way to do this, is to regard the grey area as a mask. If we repeat the experiment with the mask, only generated population points that happen to lie in the grey area are used. Since the whole of the 33% contour is grey, we will again get about 33 population points in the 33% contour. However, the 66% contour is only about 40% grey, so 60% of the cases that would fall in this area are masked out, leaving about 13 population points in the contour. Similarly, the 100% contour is only about 10% grey, so 90% of the population points will get masked out, leaving about three population points. A chi-squared test against the expected result of 33, 33, 34 will, therefore, fail. In other words, there is no point in a confidence interval including any area that cannot have produced a population point, or conversely, all points on the surface have to be able to produce a population point. The Gedankenexperiment can be extended to the situation where there are multiple "grey" regions with various probabilities of masking population points, and any number of step contours. At the limit, the mask becomes an arbitrary pdf, and the step contour approximation becomes a smooth pdf. It can be seen that the experiment is unlikely to work, if the distribution of population points is uneven. The method used in the experiment preserves the uniform distribution along the hit rate and false alarm axis, though the sorting to give a valid ROC curve produces a nonuniform, but still valid, distribution across the ROC graph surface.

VII. Simulation Results

The simulation run produced 2046 histograms in total, of which only 21 are shown here. Fig. 4 gives the histograms for the third point of four-point ROC curves over six sample sizes, and six frequencies of disease. The theoretical chi-squared distribution is plotted as a curve on each histogram so that a visual comparison can be made between the results expected in theory, and those obtained in practice. As illustrated by the diagrams, the experimental results show the expected chi-squared distributions. Given the number of cases simulated, 400 000 in each histogram, this indicates that the method is working within the limitations of the quantization of the ROC graph, into 256×256 elements and the stochastic nature of Monte Carlo simulations.

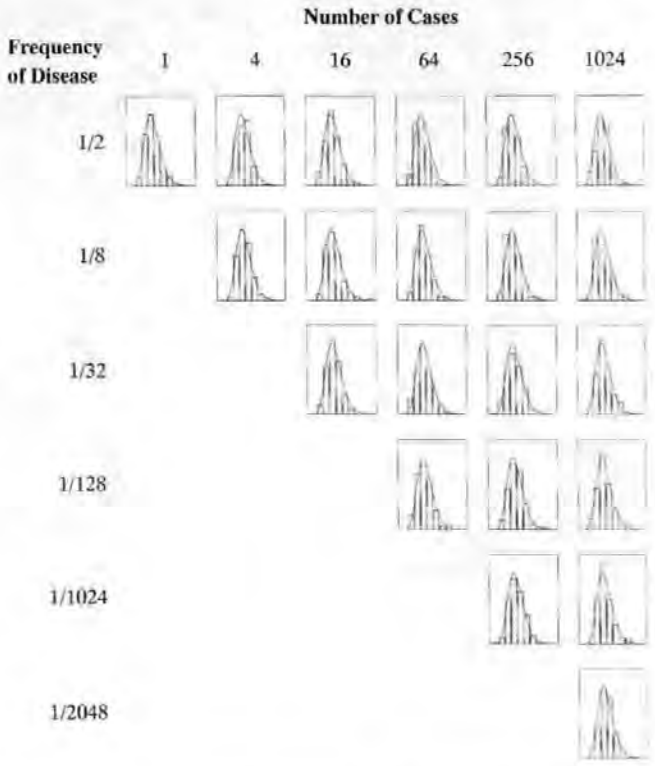


Fig. 4. Histograms of chi-squared for third point of four-point ROC curve.

VIII. Application

The method detailed above was applied to two examples of ROC analysis, published in the literature, in order to investigate the confidence boundaries produced on real data, and to illustrate how the proposed method could enhance the analysis.

The first example is taken from Swets [10]. Swets recommended the use of ROC analysis for measuring the accuracy of many types of diagnostic systems. A radiological example was presented to illustrate the use of ROC analysis, and the AUC, as the preferred single-valued measure of accuracy. A study had previously been carried out, in which six radiologists were asked to examine 118 mammograms (58 malignant, 60 benign), and classify them into one of five categories, according to likelihood that the lesion was malignant. The radiologists first diagnosed the mammograms unaided (denoted as "standard"), and then used two diagnostic aids (denoted as "enhanced"). The raw data for the pooled categorizations were given in the paper, allowing the ROC graph for the standard and enhanced diagnoses to be reproduced here as Fig. 5. From the raw data, the 95% confidence boundary of each point was calculated by the program described in Section V and these confidence boundaries are shown in Fig. 6. The entire process of calculating the ROC pdf's, and determining the 95% boundaries took approximately 5 s on a 100-MHz Pentium PC.

It should be noted that each confidence boundary is a different 2-D view of the same eight-dimensional (8-D) pdf. Each coordinate in the 8-D space represents the probability of the population

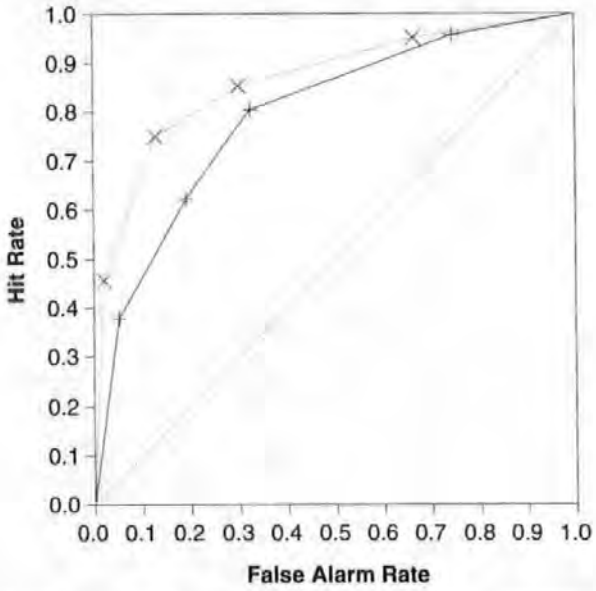


Fig. 5. ROC curve of diagnosis of 708 mammograms (from Swets [10]).

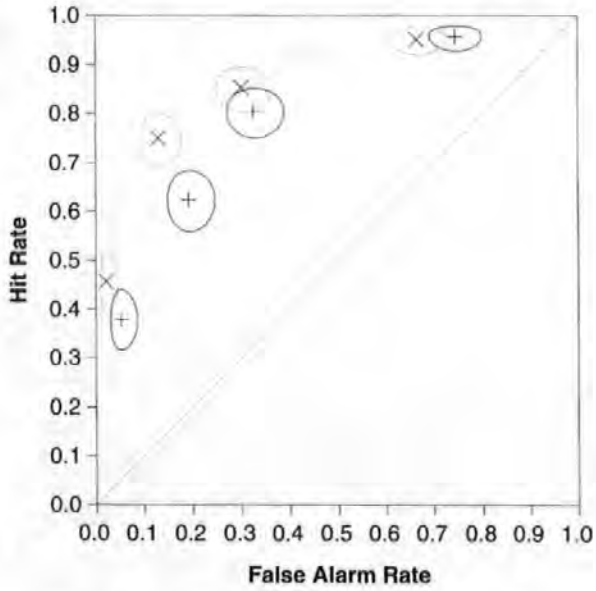


Fig. 6. The 95% confidence boundaries of ROC points in Fig. 5.

ROC curve passing through the four pairs of Hit Rate and False Alarm Rate that describe that 8-D coordinate. A 95% confidence boundary can thus be described in the 8-D hyper volume, which is the actual 95% confidence interval for the ROC curve joining the four points. This can be approximated by joining line segments through tangents to the 95% confidence boundaries of each ROC point such that the maximum and minimum areas are enclosed. It should also be noted that use of a smooth curve, or straight-line segments to join points, is an arbitrary choice outside the theory of the method. An example using straight-line segments for Swets' [10] "standard" ROC data, is shown in Fig. 7. In order to give a comparison with ROC curves in the literature, this approximation has been used to estimate the confidence interval for the AUC.

Calculating the AUC from straight line segments (Fig. 5) and the 95% CI's (Fig. 7), as described above gives nonparametric

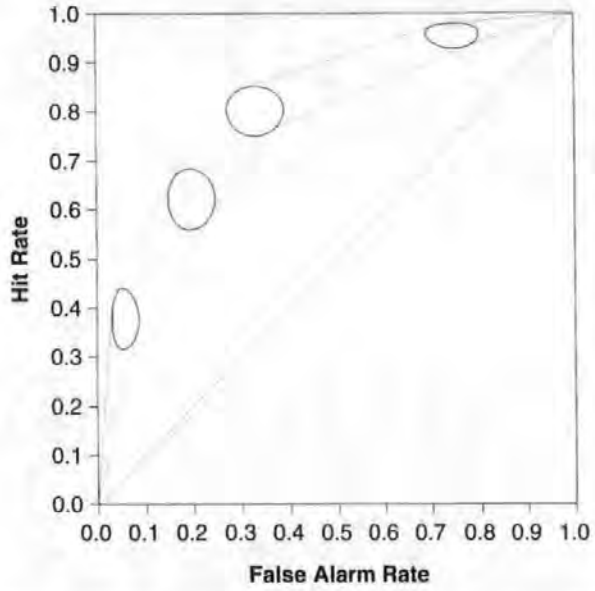


Fig. 7. The approximate 95% confidence limit of the ROC curve in Fig. 5.

values for the AUC (with 95% CI) of 0.79 (0.739–0.839) for the "standard" points of Swets [10]. Similarly, values for the AUC of 0.86 (0.815–0.898) are obtained for the "enhanced" points. Swets used a parametric method to estimate a "maximum-likelihood" curve through the points, and a corresponding parametric estimate of the AUC and its standard error. The values Swets obtained were 0.81 and 0.87, with standard errors of 0.017 and 0.014 for the "standard" and "enhanced" diagnoses respectively. Parametrically, the 95% CI is given by the mean value $\pm 2\times$ standard error, which leads to values for the AUC (with 95% CI) of 0.81 (0.776–0.844) and 0.87 (0.842–0.898). It can be seen that the nonparametric estimates obtained here, agree well with Swets' parametric estimates, albeit with slightly lower AUC's and fractionally larger confidence intervals.

The second example is taken from Adlassnig and Scheithauer [4], in which an expert system, known as CADIAG-2/PANCREAS, for the differential diagnosis of ten different types of pancreatic disease, is described. The performance of the system was compared to an histologically or clinically confirmed "gold-standard" diagnosis. Forty-seven patient records were available in which one or more of a subset of six pancreatic diseases had been diagnosed. Four patients had dual diagnoses, giving a total of fifty one diagnoses of one of six diseases. A series of ROC graphs were presented, illustrating the performance of the CADIAG-2 system in the differential diagnosis of specific diseases, both using what was described as a limited set of patient data and with the full set of available patient data. Two of Adlassnig and Scheithauer's ROC curves [4, Figures 9 and 10] illustrate the evaluations of eight diagnoses of acute pancreatitis from the 51 cases compared to the "gold-standard," using limited patient data and full patient data respectively. Although the raw data were not given, they can be accurately reconstructed from the ROC graphs, because the number of cases was small. The data are combined here and reproduced as Fig. 8. The 95% confidence boundaries calculated from the data are shown in Fig. 9. In this case, the process took only 3 s on a 100-MHz Pentium PC.

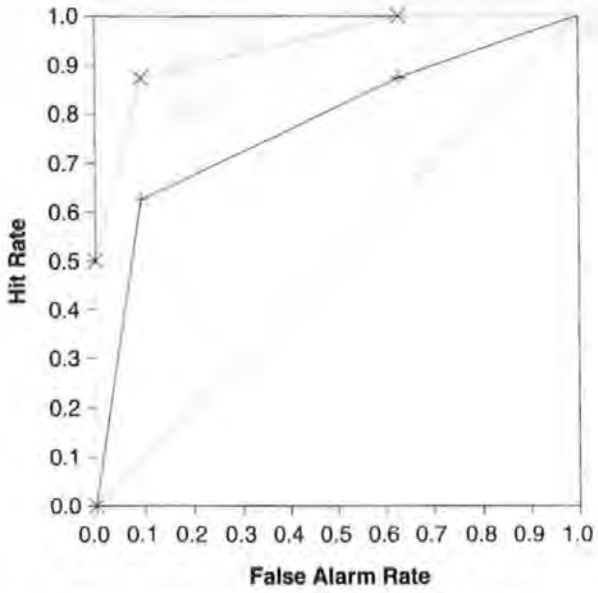


Fig. 8. ROC curve of diagnosis of acute pancreatitis from 51 cases (from Adlassnig and Scheithauer [4]).

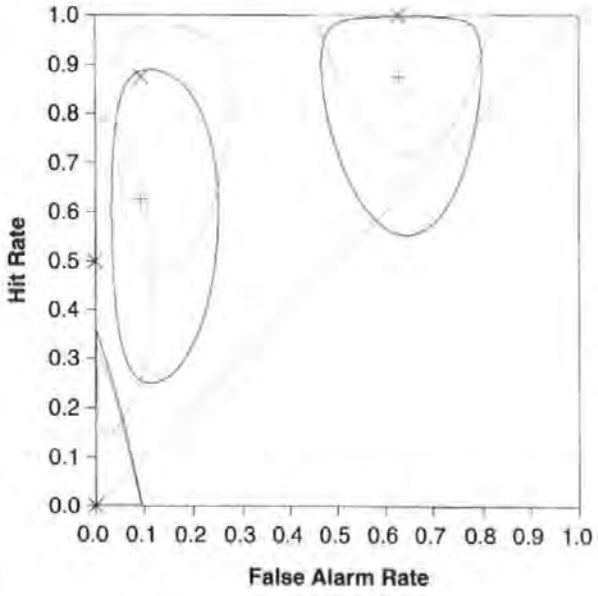


Fig. 9. The 95% confidence boundary of ROC points in Fig. 8.

Although Adlassnig and Scheithauer [4] described the use of the AUC, including testing AUC differences statistically in order to compare ROC curves, no results were given for the curves obtained. From Fig. 9, it is obvious that there is considerable uncertainty in the results due to the limited number of cases used. Using the method described above, gives an AUC of 0.79 (0.476–0.936) for the diagnoses based on "limited" patient data and 0.94 (0.617–0.981) for the diagnoses based on "full" patient data. In their analysis, Adlassnig and Scheithauer state that accuracy was always increased by adding the "full" patient data, in accordance with anticipation. While the ROC curves presented in Fig. 8 appear to support this common-sense conclusion, the large confidence boundaries in Fig. 9 suggest that this conclusion was probably premature given the data. Further, given that a random classifier has an AUC of 0.5 and a perfect classifier has an AUC of 1.0, the fact that the 95% confidence intervals of the AUC for both sets of data are so close to these extremes, illustrates how much caution should be placed in a study of such limited numbers.

Figs. 6 and 9 clearly illustrate the difference that sample size makes to the confidence that can be placed in the location of each point. While the ROC curve of the mammogram diagnoses in Fig. 5 do not look as accurate as the ROC curve for the diagnosis of acute pancreatitis in Fig. 8, examination of the confidence boundaries in Figs. 6 and 9 shows that the 708 (six opinions of 118) mammograph cases are sufficient to give good confidence of the location of the ROC points and, hence, in the curve, while the 51 pancreatic cases give a much larger confidence boundary. In particular, it can be seen by consideration of pairwise points in Fig. 6, that the points are outside of each others' confidence boundaries in all cases, and that the confidence boundaries are mutually exclusive in one case. In contrast, in Fig. 9, there is a high degree of overlap in confidence boundaries in all cases. In particular, the points with False Alarm Rate of 0.093 lie within each other's confidence boundaries, and the "limited" data point with False Alarm Rate of 0.638 lies well within the confidence boundary of the "full" point.

IX. Discussion

ROC analysis is being used with increasing popularity, in the evaluation of intelligent medical systems. If ROC curves are to be of real benefit, rather than simply being attractive drawings, the errors must be properly calculated and represented. This work establishes an important new method for generating probability distributions for all such studies.

The theoretical analysis indicates that the method derived above, is robust and accurate over any sample size, for any frequency of disease, and for any number of points. The method would appear to overcome the limitation stated by Zou *et al.* [31], that all nonparametric models are unreliable for corners of the curve since there is limited information at the extreme points. At the limit, with samples consisting of (an impractical) zero cases, the method is robust and accurate in producing a flat probability distribution over the whole surface. In other words, each point on the ROC surface is regarded as equally likely for the population *a priori*. The method is also robust with samples that do not have any false-positive or false-negative cases. Other methods [29] fail on these samples, which occur with increasing frequency, the more the diseased and healthy distributions are separated, and the few the number of cases in the sample.

This work has a further, potentially very important, application to the evaluation of intelligent medical systems. For ROC analysis to be valid, and in particular, for the AUC to be a meaningful measure, successive points on a ROC curve must be generated by altering the perceived cutoff value in the single (multivalued) output. However, in intelligent systems such as expert systems, fuzzy logic models, and artificial neural networks, internal model parameters are frequently varied or "tuned" in order to alter performance. Such alterations produce alternative outputs, which can be plotted as points on a ROC chart, but the points are not related to each other as points on a single curve. Another example would be different expert opinions of a single

fixed diagnostic test. The method presented here, allows a probability distribution to be calculated for each such point independently and hence could allow meaningful comparisons between points.

In future, extensions of the method to other aspects of ROC analysis will be investigated. First, the method will be extended to ROC curves produced by different diagnostic tests, either using paired, unpaired or partially paired data. Second, it will be extended to compare ROC curves produced by intelligent medical systems, with ROC curves produced by experts, when examining the same cases [23]. It is conjectured that it is possible to use the pdf's of ROC curves to give the exact probability that one "expert" is better than another "expert." This will improve on the present situation, where only a qualitative comparison can be made. This would be very significant, because exact risks can then be calculated for the deployment of an intelligent medical system in clinical use.

APPENDIX

To Derive (2) from (1): The *Beta* function, by definition, and given that m and n are integers, is

$$\beta(m, n) = \int_0^1 x^{m-1}(1-x)^{n-1}\,dx = \frac{(m-1)!(n-1)!}{(m+n-1)!}.$$

If $m = a_0 + 1$ and $n = a_1 + 1$

$$\int_0^1 x^{a_0+1-1}(1-x)^{a_1+1-1}\,dx = \frac{(a_0+1-1)!(a_1+1-1)!}{(a_0+1+a_1+1-1)!}$$

$$\therefore \int_0^1 x^{a_0}(1-x)^{a_1}\,dx = \frac{a_0!a_1!}{(a_0+a_1+1)!}. \tag{10}$$

Substitute (10) into (1) to give (2).

To Derive (6) from (3) and (7) from (4): The numerators of the vectors $\boldsymbol{X}$ and $\boldsymbol{Y}$ [(3), (4)] are given in terms of an expression of the following form:

$$\text{Numerator} = \int_q^r x^{a_0}(1-x)^{a_1}\,dx \tag{11}$$

where q is the probability at the lower boundary of the element, and r is the probability at the upper boundary of the element.

$$\therefore \text{Numerator} = \int_0^r x^{a_0}(1-x)^{a_1}\,dx - \int_0^q x^{a_0}(1-x)^{a_1}\,dx. \tag{12}$$

Dealing with one partial beta function at a time, where either q or r can be substituted for s

$$\int_0^s x^{a_0}(1-x)^{a_1}\,dx = \int_0^s x^{a_0}\big((1-s)+(s-x)\big)^{a_1}\,dx$$

Applying the binomial expansion

$$= \int_0^s x^{a_0}\sum_{k=0}^{a_1}\frac{a_1!}{k!(a_1-k)!}(1-s)^k(s-x)^{a_1-k}\,dx$$

$$= \sum_{k=0}^{a_1}\frac{a_1!}{k!(a_1-k)!}(1-s)^k\int_0^s x^{a_0}(s-x)^{a_1-k}\,dx. \tag{13}$$

Change the limits on the integral from zero to s, to zero to one, by letting $x = st$, which implies $dx = s\,dt$, and letting $a_2 = a_1 - k$

$$\begin{aligned}\int_0^s x^{a_0}(s-x)^{a_2}\,dx &= \int_0^1 (st)^{a_0}(s-st)^{a_2}s\,dt\\ &= \int_0^1 s^{a_0}t^{a_0}(s(1-t))^{a_2}s\,dt\\ &= \int_0^1 s^{a_0}s^{a_2}st^{a_0}(1-t)^{a_2}\,dt\\ &= s^{a_0+a_2+1}\int_0^1 t^{a_0}(1-t)^{a_2}\,dt.\end{aligned}$$

Substituting the modified Beta function as given by (10)

$$= s^{(a_0+a_2+1)}\frac{a_0!a_2!}{(a_0+a_1+1)!}. \tag{14}$$

Substituting (14) into (13)

$$\begin{aligned}&\int_0^s x^{a_0}(1-x)^{a_1}\,dx\\ &= \sum_{k=0}^{a_1}\frac{a_1!}{k!(a_1-k)!}(1-s)^k s^{a_0+a_1-k+1}\frac{a_0!(a_1-k)!}{(a_0+a_1-k+1)!}\\ &= a_0!a_1!\sum_{k=0}^{a_1}\frac{(1-s)^k s^{a_0+a_1-k+1}}{k!(a_0+a_1-k+1)!}.\end{aligned} \tag{15}$$

Substituting (15) into (12) and then substituting into (11) and simplifying

$$\text{Numerator} = a_0!a_1!\sum_{k=0}^{a_1}\frac{r^{a_0+a_1+1-k}(1-r)^k - q^{a_0+a_1+1-k}(1-q)^k}{k!(a_0+a_1+1-k)!}. \tag{16}$$

Substituting (16) into (3) gives the expression for each element of the $\boldsymbol{X}$ vector, as shown in the equation at the top of the next of the page, which simplifies to (6).

Similarly, for $\boldsymbol{Y}$ [by substituting (16) into (4) and simplifying to give (7)].

To Derive (9) From (8): Substituting (8) into (14), where s is, in turn, $1 - s_0 - x_1 - x_2$, $1 - s_0 - x_1$, and $1 - s_0$

$$\begin{aligned}f(s_0) &= \int_0^{1-s_0}\int_0^{1-s_0-x_1} s_0^{a_0}x_1^{a_1}x_2^{a_2}\\ &\quad\cdot(1-s_0-x_1-x_2)^{a_3+a_4+1}\frac{a_3!a_4!}{(a_3+a_4+1)!}\,dx_2\,dx_1\\ &= \int_0^{1-s_0} s_0^{a_0}x_1^{a_1}(1-s_0-x_1)^{a_2+a_3+a_4+2}\\ &\quad\cdot\frac{a_2!(a_3+a_4+1)!}{(a_2+a_3+a_4+2)!}\frac{a_3!a_4!}{(a_3+a_4+1)!}\,dx_1\\ &= s_0^{a_0}(1-s_0)^{a_1+a_2+a_3+a_4+3}\\ &\quad\cdot\frac{a_1!(a_2+a_3+a_4+2)!}{(a_1+a_2+a_3+a_4+3)!}\frac{a_2!a_3!a_4!}{(a_2+a_3+a_4+2)!}.\end{aligned}$$

Simplifying, gives (9).

$$X_i = \frac{a_0!a_1!}{\frac{a_0!a_1!}{(a_0+a_1+1)!}} \sum_{k=0}^{a_1} \frac{\left(\frac{i}{n}\right)^{a_0+a_1+1-k}\left(1-\frac{i}{n}\right)^{k} - \left(\frac{i-1}{n}\right)^{a_0+a_1+1-k}\left(1-\frac{i-1}{n}\right)^{k}}{k!(a_0+a_1+1-k)!}$$

For all i from $i = 1$ to $i = n$

ACKNOWLEDGMENT

The authors would like to thank J. Stander for advice on the presentation of the results of the Monte Carlo simulation and for his editorial suggestions.

REFERENCES

[1] J. M. Garibaldi and E. C. Ifeachor, "Application of simulated annealing fuzzy model tuning to umbilical cord acid-base interpretation," *IEEE Trans. Fuzzy Syst.*, vol. 7, pp. 72–84, Jan. 1999.

[2] J. W. Huang, Y. Lu, A. Nayak, and R. J. Roy, "Depth of anesthesia estimation and control," *IEEE Trans. Biomed. Eng.*, vol. 46, pp. 71–81, Jan. 1999.

[3] D. A. Cairns, J. H. L. Hansen, and J. E. Riski, "A noninvasive technique for detecting hypernasal speech using a nonlinear operator," *IEEE Trans. Biomed. Eng.*, vol. 43, pp. 35–45, Jan. 1996.

[4] K. P. Adlassnig and W. Scheithauer, "Performance evaluation of medical expert systems using ROC curves," *Comput. Biomed. Res.*, vol. 22, no. 4, pp. 297–313, 1989.

[5] L. G. Koss, M. E. Sherman, M. B. Cohen, A. R. Anes, T. M. Darragh, L. B. Lemos, B. J. McClellan, D. L. Rosenthal, S. Keyhani-Rofagha, K. Schreiber, and P. T. Valente, "Significant reduction in the rate of false negative cervical smears with neural network-based technology (PAPNET Testing System)," *Human Pathol.*, vol. 28, no. 10, pp. 1196–1203, 1997.

[6] S. Andreassen, A. Rosenfalck, B. Falck, K. G. Olesen, and S. K. Andersen, "Evaluation of the diagnostic performance of the expert EMG assistant MUNIN," *Electroencephalogr. Clin. Neurophysiol.*, vol. 101, pp. 129–144, 1996.

[7] B. V. Ambrosiadou, D. G. Goulis, and C. Pappas, "Clinical evaluation of the DIABETES expert system for decision support by multiple regimen insulin dose adjustment," *Comput. Meth. Programs Biomed.*, vol. 49, pp. 105–115, 1996.

[8] H. D. Cheng, Y. M. Lui, and R. I. Freimanis, "A novel approach to microcalcification detection using fuzzy logic technique," *IEEE Trans. Med. Imag.*, vol. 17, no. 3, pp. 442–450, 1998.

[9] R. D. F. Keith, S. Beckley, J. M. Garibaldi, J. A. Westgate, E. C. Ifeachor, and K. R. Green, "A multicentre comparative study of 17 experts and an intelligent computer system for managing labor using the cardiotocogram," *Br. J. Obstetrics Gynaecol.*, vol. 102, pp. 688–700, 1995.

[10] J. A. Swets, "Measuring the accuracy of diagnostic systems," *Science*, vol. 240, pp. 1285–1293, 1988.

[11] K. Clarke, R. O'Moore, R. Smeets, J. Talmon, J. Brender, P. McNair, P. Nykanen, J. Grimson, and B. Barber, "A methodology for evaluation of knowledge-based systems in medicine," *Artif. Intell. Med.*, vol. 6, pp. 107–121, 1994.

[12] R. Engelbrecht, A. Rector, and W. Moser, "Verification and validation," in *Assessment and Evaluation of Information Technologies*, E. M. S. J. van Gennip and J. L. Talmon, Eds. Amsterdam, The Netherlands: IOS, 1995, pp. 51–66.

[13] A. R. Henderson, "Assessment of clinical enzyme methodology: a probabilistic approach," *Clinica. Chimica. Acta*, vol. 257, pp. 25–40, 1997.

[14] A. A. Renshaw, K. R. Lee, and S. R. Granter, "Use of statistical analysis of cytologic interpretation to determine the causes of interobserver disagreement and in quality improvement," *Cancer Cytopathol.*, vol. 81, no. 4, pp. 212–219, 1997.

[15] S. Y. Jang, R. J. Jaszczak, B. M. W. Tsui, C. E. Metz, D. R. Gilland, T. G. Turkington, and R. E. Colman, "ROC evaluation of SPECT myocardial lesion detectability with and without single iteration nonuniform Chang attenuation compensation using an anthropomorphic female phantom," *IEEE Trans. Nucl. Sci.*, pt. 2, vol. 45, pp. 2080–2088, Aug. 1998.

[16] D. J. Goodenough, K. Rossmann, and L. B. Lusted, "Radiographic applications of signal detection theory," *Radiology*, vol. 105, pp. 199–200, 1972.

[17] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve," *Diagnostic Radiol.*, vol. 143, no. 1, pp. 29–36, 1982.

[18] B. J. McNeil and J. A. Hanley, "Statistical approaches to the analysis of receiver operating characteristic (ROC) curves," *Med. Decision Making*, vol. 4, pp. 137–150, 1984.

[19] S. C. Kazmierczak, P. G. Catrou, and F. Van Lente, "Diagnostic accuracy of pancreatic enzymes evaluated by use of multivariate data analysis," *Clinical Chem.*, vol. 39, no. 9, pp. 1960–1965, 1993.

[20] L. Ohno-Machado, "A comparison of Cox proportional hazards and artificial neural network models for medical prognosis," *Comput. Biol. Med.*, vol. 27, no. 1, pp. 55–65, 1997.

[21] D. Bamber, "The area above the ordinal dominance graph and the area below the receiver operating graph," *J. Math. Psychol.*, vol. 12, pp. 387–415, 1975.

[22] N. A. Obuchowski and M. L. Lieber, "Confidence intervals for the receiver operating characteristic area in studies with small samples," *Academic Radiol.*, vol. 5, pp. 561–571, 1998.

[23] C. E. Metz, "Statistical analysis of ROC data in evaluating diagnostic performance," in *Multiple Regression Analysis: Applications in the Health Sciences*, D. Herbert and R. Myers, Eds. New York: American Institute of Physics, 1986, pp. 365–384.

[24] E. R. DeLong, D. M. DeLong, and D. L. Clarke-Pearson, "Comparing the areas under two or more correlated receiver operating characteristic curves: A nonparametric approach," *Biometrics*, vol. 44, pp. 837–845, 1988.

[25] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*. New York, NY: Chapman & Hall, 1993.

[26] S. W. Greenhouse and N. Mantel, "The evaluation of diagnostic tests," *Biometrics*, vol. 6, pp. 399–412, 1950.

[27] H. Schafer, "Efficient confidence bounds for ROC curves," *Statistics Med.*, vol. 3, pp. 1551–1561, 1994.

[28] R. A. Hilgers, "Distribution-free confidence bounds for ROC curves," *Meth. Inform. Med.*, vol. 30, pp. 137–150, 1991.

[29] C. E. Metz, B. A. Herman, and J. Shen, "Maximum likelihood estimation of receiver operating characteristic (ROC) curves from continuously-distributed data," *Statistics Med.*, vol. 17, pp. 1033–1053, 1998.

[30] P. L'Ecuyer, "Efficient and portable combined random number generators," *Commun. ACM*, vol. 31, no. 6, pp. 742–774, 1988.

[31] K. H. Zou, W. J. Hall, and D. E. Shapiro, "Smooth nonparametric receiver operating characteristic (ROC) curves for continuous diagnostic tests," *Statistics Med.*, vol. 16, pp. 2143–2156, 1997.

Julian B. Tilbury received the HND degree from University of Plymouth (formerly Plymouth Polytechnic), Devon, U.K., in 1984. He received the M.Sc. degree in intelligent systems and cognitive psychology from Plymouth University in 1996. He is currently a Ph.D. degree student in the Center for Intelligent Systems, University of Plymouth.

He was a Computer Programmer at the Electricity Council Research Center, Capenhurst, the Computer Center, Salford University, and Salford Software for 11 years after receiving the HND degree and before returing to his studies at Plymouth.

Peter W. J. Van Eetvelt B.Sc.(Eng), B.A., M.Sc., C.Eng., MIEE, C. Math., FIMA received a second-class upper honors degree in electrical engineering from Brighton University, Sussex U.K., in 1965, the first–class honors degree in mathematics and the M.S. degree in applied mathematics from the Open University, Milton Keynes, U.K., in 1981 and 1987, respectively.

He spent most of his career in aerospace and defence industries as a Research, Design, and Development Engineer and Applied Mathematician. In 1988, he was appointed as a Senior Lecturer at Plymouth University, Plymouth, U.K., where he contributed his mathematical expertise to several papers covering a wide range of disciplines and conducted fundamental research into peak factor reduction problem for OFDM transmission schemes. He is currently a Consultant whose main interest is in the application of mathematical methods to engineering problems.

Jonathan M. Garibaldi received the B.Sc. (Honors) degree in physics from Bristol University, Bristol, U.K., and the M.Sc. (intelligent systems) and Ph.D. degrees from The University of Plymouth, Devon, U.K., in 1984, 1991, and 1997, respectively. He entered the University of Plymouth in 1990 to work toward the M.Sc. degree. After a brief return to the software industry he re-entered the University of Plymouth as a Research Assistant in 1992 (School of Electronics, Communication and Electrical Engineer) and subsequently went on to receive the Ph.D. degree in medical uncertainty.

After receiving the B.Sc. degree in 1984, he spent six years in industry as a Software Engineer. He continued at the University of Plymouth as a Research Fellow working on data mining and intelligent medical systems. In September 1999, he became a Senior Lecturer in Computer Science in the Department of Computer Science, Faculty of Computer Sciences and Engineering, De Montfort University, Leicester, U.K. He is a member of the Center for Computational Intelligence specializing in fuzzy applications of intelligent medical systems. He also has interests in general intelligent systems techniques, data mining, probability, statistics, and computability.

John S. H. Curnow received the Honors degree in electrical and electronic engineering from the University of Bath U.K., in 1967.

He became a member of the Institute of Electrical Engineers in 1982 and Fellowship of the Institute of Physicists and Engineers in Medicine in 1988. He has worked in the field of medical electronics since 1968 and his present interests are in developing new areas of clinical patient monitoring.

Emmanuel C. Ifeachor received the B.Sc. (Honors) degree in communication engineering from the University of Plymouth (formerly Plymouth Polytechnic), Devon, U.K., in 1980, the M.Sc. degree in communication engineering, the Diploma of membership of Imperial College (DIC) from Imperial College, London, U.K., in 1981, and the Ph.D. degree in medical electronics from University of Plymouth in 1985.

He served as Head of School of Electronic, Communication and Electrical Engineering from 1995 to 1999. He is currently a Professor of Intelligent Electronic Systems and Head of the newly created Center for Communications, Networks, and Information Systems at the University of Plymouth. His main research interests are signal processing and intelligent systems with applications in biomedicine, audio and telecommunications. He has successfully led many industry- and Government-funded projects in these areas. He has published over 100 technical papers, co-authored *Digital Signal Processing—A Practical Approach* (Reading, MA: Addison Wesley, 1993), and edited or co-edited six books in signal processing and intelligent systems, including *Artificial Neural Networks for Biomedicine* (London, U.K.: Springer, 2000).

Dr. Ifeachor recently received the Institute of Electrical Engineers Dr. V. K. Zworykin Premium, twice, for his work on fetal electrocardiogram analysis and EEG analysis. He currently serves on the Institute's Professional Group on Medical Focus and on the committee for U.K. Professors and Heads of Electrical Engineers.